Industrial Applications of Data Mining

Engineering Effort Forecasting based on Mining and Analysis of Patterns in Historical Project Execution Data

by

Indrani Bhattacharya

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved January 2013 by the
Graduate Supervisory Committee:

Arunabha Sen, Co-Chair
Karl Kempf, Co-Chair
Huan Liu

ARIZONA STATE UNIVERSITY

May 2013

ABSTRACT

Data mining is increasing in importance in solving a variety of industry problems. Our initiative involves the estimation of resource requirements by skill set for future projects by mining and analyzing actual resource consumption data from past projects in the semiconductor industry. To achieve this goal we face difficulties like data with relevant consumption information but stored in different format and insufficient data about project attributes to interpret consumption data. Our first goal is to clean the historical data and organize it into meaningful structures for analysis. Once the preprocessing on data is completed, different data mining techniques like clustering is applied to find projects which involve resources of similar skillsets and which involve similar complexities and size. This results in "resource utilization templates" for groups of related projects from a resource consumption perspective. Then project characteristics are identified which generate this diversity in headcounts and skillsets. These characteristics are not currently contained in the data base and are elicited from the managers of historical projects. This represents an opportunity to improve the usefulness of the data collection system for the future. The ultimate goal is to match the product technical features with the resource requirement for projects in the past as a model to forecast resource requirements by skill set for future projects. The forecasting model is developed using linear regression with cross validation of the training data as the past project execution are relatively few in number. Acceptable levels of forecast accuracy are achieved relative to human experts' results and the tool is applied to forecast some future projects' resource demand.

ACKNOWLEDGMENTS

I would like to acknowledge and thank my thesis mentor, Karl Kempf and my graduate advisor, Arunabha Sen who were very helpful and offered invaluable guidance, assistance and support. I would also like to thank my thesis committee member, Huan Liu for his advice.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1. Business Problem

Resource planning and utilization in industries has always been a challenge. It has gained even more importance in today's competitive world. Inefficient planning directly impacts the revenue, quality and quantity of the deliverables. Competitors gain more advantage as the reputation of the company is affected by inefficient planning. Therefore, accurate engineering resource allocation and efficient use of the resources is paramount for smooth execution of future projects.

The research questions on above business problem are:

a) Can we extract necessary information about the resource requirements for future projects by analyzing past project execution data ("Project Actuals")?

b) Is there a way to prevent past mistakes in planning being repeated in future projects?

High Level Objective: Develop a decision-support system by using relevant information from past project data for future resource planning and utilization, thereby significantly reducing human errors in the planning process.

The above research goal can be refined in the following way:

- Objective 1: Analyze past project execution data and ensure its quality. The data should contain a good representation of resource consumption over time for different skillsets of a project or a group of similar projects.
- Objective 2: Provide users a method to compare different projects in the same family (category) or same complexity in terms of resource consumption in different functional areas. This helps in selecting good candidates of data to be used in future forecasting.
- Objective 3: Forecast the future resource requirements for a project/product.

The resource forecasting problem takes different forms and encounters numerous challenges when applied to various domains. The current thesis endeavors to estimate resource requirement for future projects by mining and analyzing actual resource consumption data from past projects in the Semiconductor industry. In order to validate the core research premise of resource utilization and planning, the thesis has used past project data from Intel Corporation. Intel is the world's largest Semiconductor chip manufacturing company and it has tens of product categories in server, desktop and mobile markets. Each Intel product is produced over multiple generations and each product of each generation is the result of execution of complex projects carried out over a period of time. During the resource planning phase, multiple aspects of the projects such as (i) skillset required, (ii) for how long, (iii) in what order and (iv) magnitude of headcount in each skills at different point of time, must be considered.

At Intel, each such project requires involvement of a number of design engineers (front end), manufacturing engineers and technicians (back end), validation engineers, platform engineers and firmware/software engineers. The projects can be different in the following ways.

- The order in which different engineering skills are required can vary over project categories. For example, server central processing unit (CPU) projects require early involvement of a large number of circuit design engineers while platform board projects do not need any circuit design engineers. Rather, they need a large number of validation engineers.
- There is always a variation of resource headcounts in magnitude and duration in projects belonging to the same technical category. We elaborate this point with an example. Suppose two projects A and B belong to the same project category, but their reuse of design and/or components from a past project C are quite different. While project A significantly reuses design or components from C, project B does not. In this scenario, design engineer requirements for A will be

significantly smaller than that of B, while the number of validation engineers may remain same.

- Another challenge is in tracking skillsets using the timesheet. For instance 150 different skillsets might manifest in a timesheet. However planning for resources at this fine level of granularity is extremely complex. At the same time, aggregating all skills into one group will result in an overly coarse grain categorization which may be of little use. As such the challenge is to find the right level of granularity for categorization that can be utilized for meaningful forecasting.

This thesis makes an effort to solve the above challenges in resource forecasting using data mining of historical data.

1.2. Data Mining in Industry Application

Data mining is a technique to analyze data from different perspectives and dimensions, identify patterns and relations and discover useful information. It has gained immense importance in solving a variety of industry problems. As studied here the information is used to predict future behavior with higher accuracy. Some popular methods of data mining are the following:

- Classification—This is a process of categorizing the new data based on the patterns of old data,
- Clustering— Grouping of data which are similar in some perspective. This is one of the example of unsupervised learning,
- Association—Determine the likelihood of co-occurrences of data in the future based on their past behavior,
- Regression— Describes data patterns by mathematical functions which may be linear or non-linear.

These methods are effective when there is a fairly large volume of high quality data available for mining. However in the real world data mining becomes more challenging because of the following reasons.

- Unavailability of a sufficiently reliable and complete dataset
- Redundancy in available data
- Unavailability of data in a uniform format

These situations are encountered either individually or in combination making applied data mining in industry much different from classical data mining. This thesis overcomes such data challenges encountered in the Semiconductor industry (described in detail in chapter 3).

1.3. Contribution

This thesis provides a model for resource demand forecasting by mining and analyzing past projects execution data in the Semiconductor industry. The following are the key contributions of this thesis:

- It presents a forecasting model which generates bias-free resource demand predictions (Details are in chapters 6 and 7).
- It addresses issues related to insufficient, unreliable and incomplete data sets often encountered in the Semiconductor industry and it provides guideline on i) what data to collect and store and ii) how to effectively utilize such data to solve broader industrial problems (Details are in chapter 3).
- It examines whether resource forecasting methods developed in other domains can be utilized in Semiconductor domain.
- It also demonstrates that it is not possible to generate a single resource requirement template for each product category (Details are in chapter 4).
- It identifies the features that impact resource forecasting by developing a semi-supervised feature selection approach (A mixture of rules from experts and past data analysis) (Details are in chapters 5 and 6).

Chapter 2

LITERATURE REVIEW AND RELATED WORK

Accurate human resource forecasting and staffing planning is important and has always been a critical problem in industries. In the last 10 years this decision problem has gained prominence and has emerged as an interesting research area. There are many publications that deal with optimal human resource forecasting in which different research groups used various approaches, methods and algorithms to achieve the same goal. However, the approaches vary depending on business requirements and resource forecasting problems take various forms based on different business domains in which they are applied.

In References [6, 7], the authors dealt with optimization of human resource utilization in the service engagement domain by applying integer programming (IP). In service systems there is always a demand and supply of resources of different skillsets and different levels of expertise. For every successful match of a demand position and supply position based on skills and experience level they generate a success score while for every unassigned resource or unattended project they generate a penalty score. Using these two types of scores they generated a utility function and the objective of using IP is to maximize the utility function within some domain specific constraints. Moreover the demand for resources is generated from predefined templates of resource utilization for project categories. New or future project names are matched with the old projects to identify their categories using cosine similarity or pairwise distance measure giving importance on keyword match over common word match (TF/IDF – Term Frequency and Inverse Document Frequency).

A similar approach using mixed integer programming (MIP) is described in [5] where the selection of projects, schedules and resources in a service system is optimized under budget constraints. In [4] a similar IP approach is applied on product development projects as opposed to service engagement projects.

In [3] the authors solved this problem for the service engagement projects by applying the data mining method of clustering to group projects which have similar resource utilization profiles involving the same skills. It also describes how to develop a project taxonomy based on the cluster profiles. It is a model based sequence clustering approach where each sequence is a project resource distribution over time (e.g. weekly). Clustering is based on a Hidden Markov Model.

The publications most closely related to this thesis are [1] and [2]. In [1] the resource forecasting for service engagement projects is done using a statistical data mining approach. The service projects having similar resource utilization profile over time for different skills (headcount per skill/total heads * 100) are clustered and one template is generated per cluster describing the resource distribution for that project family. The total numbers of heads are obtained by applying a regression model on expected revenue as revenue and total labor hour are highly correlated in service engagement projects. Distribution of resources for project categories is applied to total heads to get the actual headcounts per skill. [2] is an extension of [1] where the category of the future projects are assigned by applying keyword based matching technique with previous projects (cosine similarity in project names). Moreover, a better estimation is obtained by applying semi-supervised learning such as soft-seeded K means clustering on project data.

All of the above approaches are effective in solving different instantiations of human resource forecasting problems in different domains. However, when we select the Semiconductor industry as a domain, the human resource demand forecasting has to be done for product development projects. Differences between Service Engagement projects and product based Semiconductor projects include:

- In service engagement projects the cost of the project is mainly the human resource cost. The revenue is the cost plus a generated markup for profit. Unlike service engagement projects, in a Semiconductor industry or any product based company, the cost of the projects is dependent on many factors in addition to

human resources and the revenue depends on the sales of the product. To meet the demand and the customer's expectations and to take a good position in competitive markets, technical product manufacturers have to introduce attractive technical features periodically in different releases. For example in a Server/Client CPU the features can be more speed, more memory, lower power, lower cost, lower heat, reduced size, etc., altogether called RAS features (Reliability, Availability and Serviceability) . The resource requirements depend mainly on the type of project/ product/component and its complexity.

- In service engagement projects it is assumed that the percentage distribution of efforts in different skillsets is the same in all projects of the same group in the project taxonomy. In the Semiconductor industry the resource requirement pattern for different generations of the same product changes not only in terms of total efforts but also in terms of the efforts in different skillset (Explained in chapter 4).
- Unlike service projects, Semiconductor projects use code names to maintain their roadmap secrecy. So the categories of future projects are very difficult to be assigned by comparing project names. Rather technical feature metrics can be used.

This thesis endeavors to apply the data mining techniques which proved effective for service engagement projects to product oriented projects especially in the Semiconductor industry to make a better resource forecast model dependent on past project resource utilization patterns and project complexities derived from technical features of the products. (Details are in chapter 4, 5 & 6)

References [8, 15] helped in getting an idea about best practices in data mining and pros and cons of different methodologies which in turn helped in selecting appropriate methods. Reference [16] is used as a guide for regression (data mining method).

Another aspect of this thesis is to make the available data useful and minimize the effect of missing/incomplete data. There are several papers published on preprocessing techniques of data mining such as data cleaning and data transformation. In [10, 14] the authors described various methods to handle data quality issues for example, missing values, integrity constraint violation, unique constraint violation and bringing data in various format into a uniform format. The major challenge was faced in making sense of the data though the data maintained all sorts of constraints applied to a successful data-warehouse. Another data challenge is handling the growing size of data. The raw data source is basically the timesheet data for individual employees per project per month. The data needs to be aggregated to a higher level based on a mining perspective for the ease of mining and to reduce data volume. Reference [9] describes diverse OLAP techniques for data analytics/summarization. Those data centric challenges are discussed in detail in chapter 3.

Chapter 3

DATA ACQUISITION AND TRANSFORMATION RELATED CHALLENGES AND SOLUTION

3.1 Data Description:

The data used for this analysis is execution data of past Semiconductor projects referred as "Actuals". Figure 1 illustrates the star schema of the data warehouse storing past project data. The efforts expended by each employee in each project for a month are recorded in the database fact table. This data is obtained from employees' weekly timesheet data. The efforts recorded in the fact table can be aggregated to different dimensions as mentioned below – (The hierarchies are also shown in Figure 1)

a) Time Hierarchy

b) Skillset Hierarchy

c) Organizational Unit Hierarchy

d) Project/ Program/ Product Line (PPP) Hierarchy

e) Site Hierarchy

For better analysis, the first step was to aggregate the data at some level higher than the individual employee level. The purpose of this higher level data aggregation is to extract the information about the number of person-quarters effort required to complete a particular task/ project/ program (PPP) per skillset.

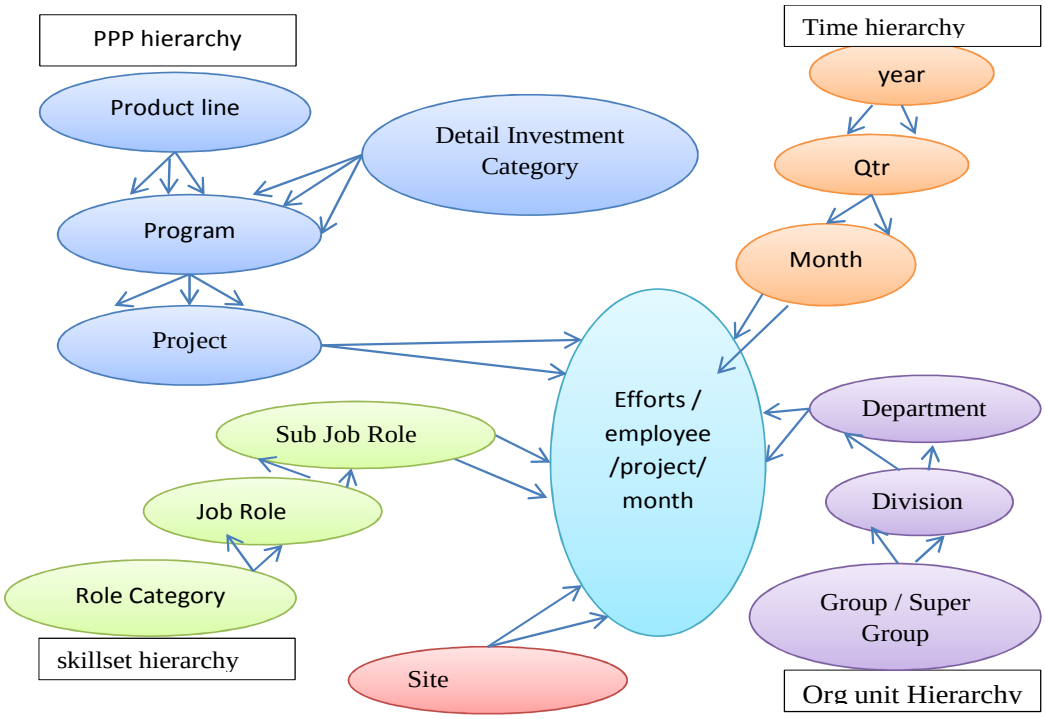


Figure 1. An Illustration of Data Warehouse Star Schema for Project Actuals

3.2 Data Challenges and Solutions

Though the data warehouse in Figure 1 maintains all sorts of integral, referential constraints at the database level to ensure high data quality, unfortunately there are situations when it becomes difficult to make sense of the data or extract information from the data. The challenges include:

3.2.1 ***Data unavailability or incompleteness in the data due to lack of 100% compliance in the project actuals source system***

- **Type 1 data challenge**: The design, development, manufacturing and validation work for a program or project involves multiple divisions in different geographical locations across the multinational organization. Unfortunately not all of the divisions achieved 100% compliance in registering project actuals on a single day. The points in Figure 2 show the quarter wise full time equivalent (FTE) (head counts) resource consumption for completing a validation job of a particular program. It is unlikely that the headcounts will reach its

maximum (near 200) starting from 0 in one quarter. This is an example of timesheet data entry non-compliance from quarter 5-10.

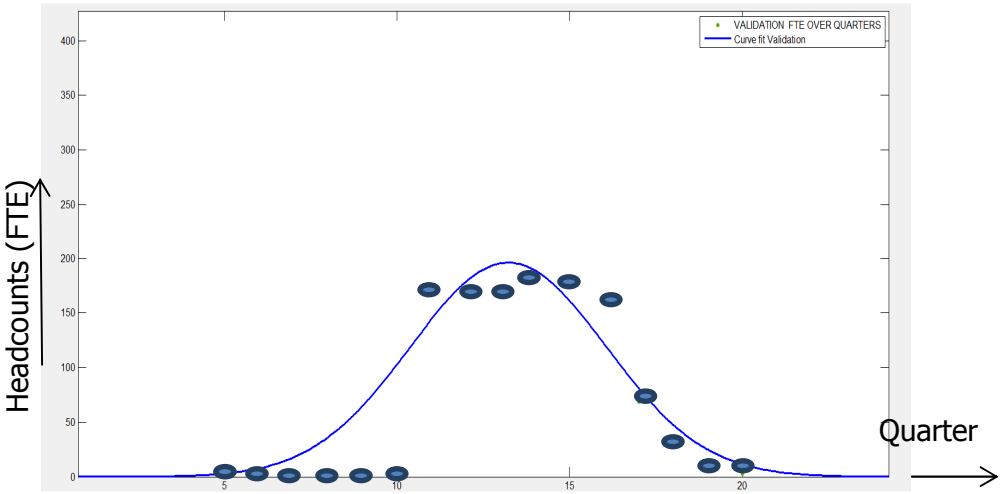


Figure 2: Illustrates the data incompleteness type 1 due to human error

A good way to manage this efficiently is to check the resource consumption profiles for programs of similar product family. For the above example in Figure 2,

- Observe the engagement of validation resources with other skillsets like whether validation engineers are involved from the very beginning of the projects or few quarters after the design work started or something else.
- Observe the graphical representation of other functional areas of that project and figure out the start time/quarter for misreported "validation" work.
- Then apply curve fitting for the misreported part. (Curve fitting strategies are explained in detail in later part of this chapter)

The line in Figure 2 shows the curve fit for the data points and the curve between quarters 5-10 will fill the gap of missing data for the project.

- **Type 2 Data Challenge:** Another form of this problem is the headcount is abnormally low for a particular quarter. For example, in Figure 3 quarter 12 has a lower head count reporting than its previous as well as next quarters which is not a desirable scenario and

violates the project curve pattern. Failure to meet 100% compliance for one quarter or two gives rise to spikes in the curve which needs smoothing for better estimation.

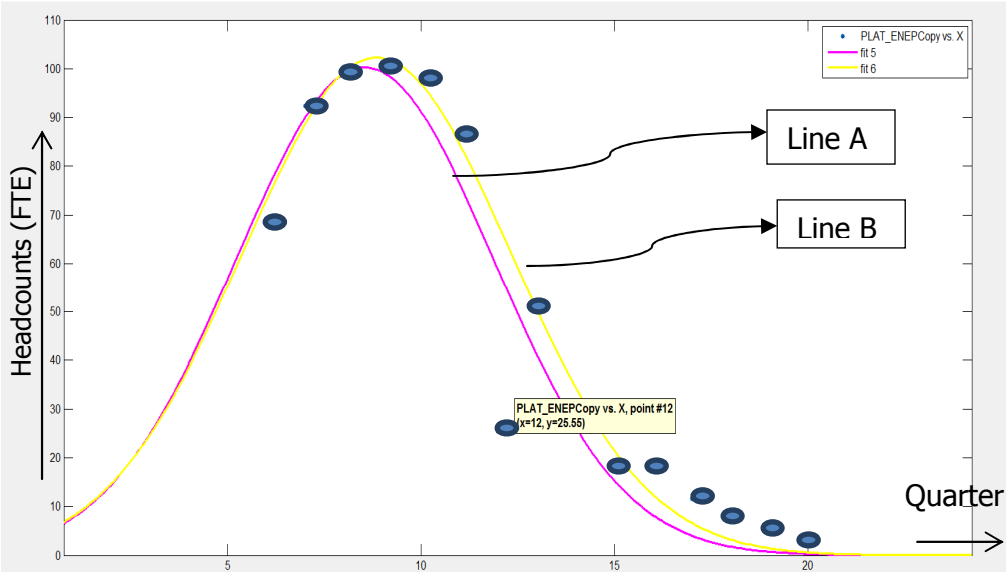


Figure 3: Illustrates the data incompleteness type 2 due to human error

In Figure 3, line A shows the curve fit plot using the data point for quarter 12 which has incomplete data and line B shows the curve fit after smoothing the data. Their difference in area under the curve shows the error occurred due to the consideration of this "bad" data point.

- **Type 3 Data Challenge:** Another version of this problem is partial compliance for a long period which is even more difficult to be identified and resolved.

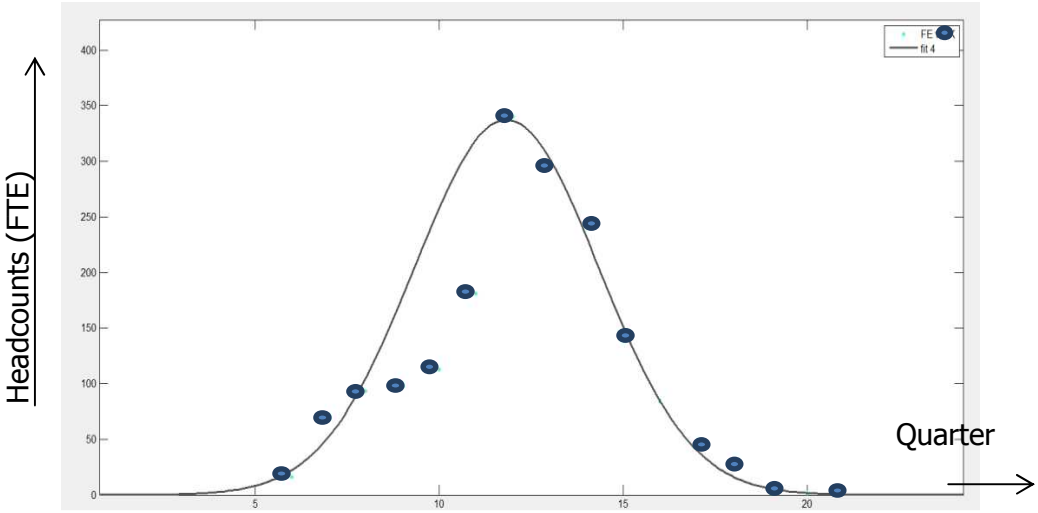


Figure 4: Illustrates the data incompleteness type 3 due to human error

In Figure 4 the resource consumption for design efforts have been plotted against quarters and it is obvious from the data points that there were some anomalies in the data from quarter 6-11. The interesting question is whether that project was really on hold over 2-3 quarters with the same headcounts or is this effect of partial compliance to time sheet data from quarter 6-11?

To resolve this confusion the approach is to observe the other functional areas of the project. For example, if design headcounts show some unlikely pattern, check the manufacturing, validation and platform work profiles for that project. (The rationale behind selecting these role categories as standard is described in last part of chapter 3 and in chapter 4). If all of the functional areas show that there was a duration for which each one maintained a constant heads of resources then obviously there was a hold on that project. If only one functional area shows anomalies then probably it is a case of misreporting / absence of reporting actuals.

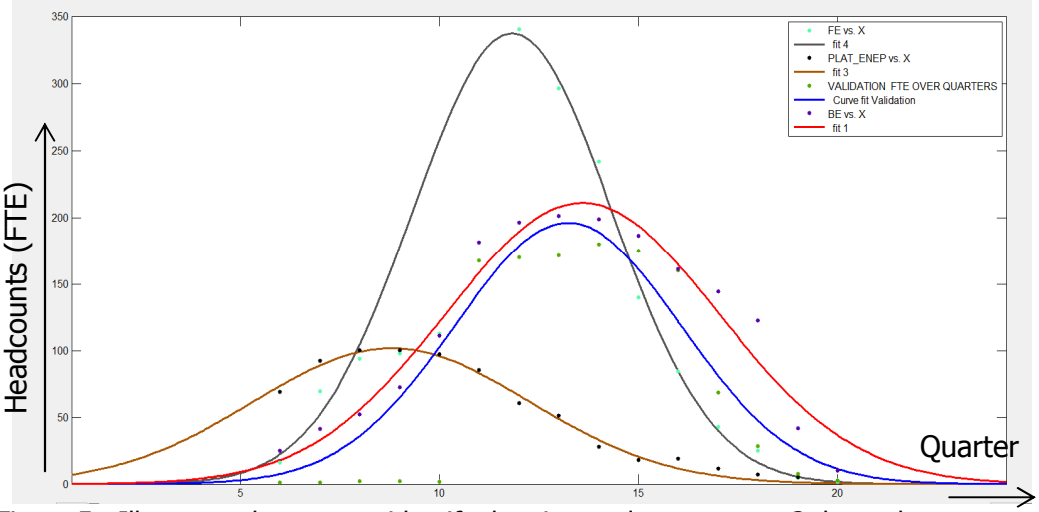


Figure 5 : Illustrates the ways to identify data incompleteness type 3 due to human error

Figure 5 confirms that there was a partial reporting for design heads because the other functional areas maintained constant/uniform increasing slope in the same duration.

3.2.2 Non standardization in nomenclature of projects / programs across different divisions:

The same project/program can be called by different names in different divisions throughout the project or their names can vary quarter to quarter within the same division.

Examples:

1) The same program is called Phoenix and PHX in different divisions though they both refer to same program/project.

2) The same program is called by different names like Phoenix and PHX in different quarters within same division.

3.2.3 Non standardization in nomenclature of skillsets across different divisions:

The same skillsets can be called by different names in different divisions throughout the project or their names can vary quarter to quarter within the same division.

Examples:

1) The same skillset is called "Design Engineer" and "Des Engg" in different divisions though they both refer to same job role.

2) The same skillset is called by different names like "Design Engineer" and "Des Engg" in different quarters within same division.

3.2.4 Effect of Major Reorganization: Merging / splitting of skillsets/ divisions/ projects/ programs / combination of all.

Due to reorganization, an employee might have multiple labels (designations) assigned to him/her at different points in time during a project cycle. This results in non-standardization in nomenclature and prevents the data from being used in broader level analysis.

Case 1: Splitting of skillsets: One of the simple effects of reorganization is splitting of skillsets. Though resources continued to do same work over subsequent quarters, their level of reporting in actuals became different in granularity.

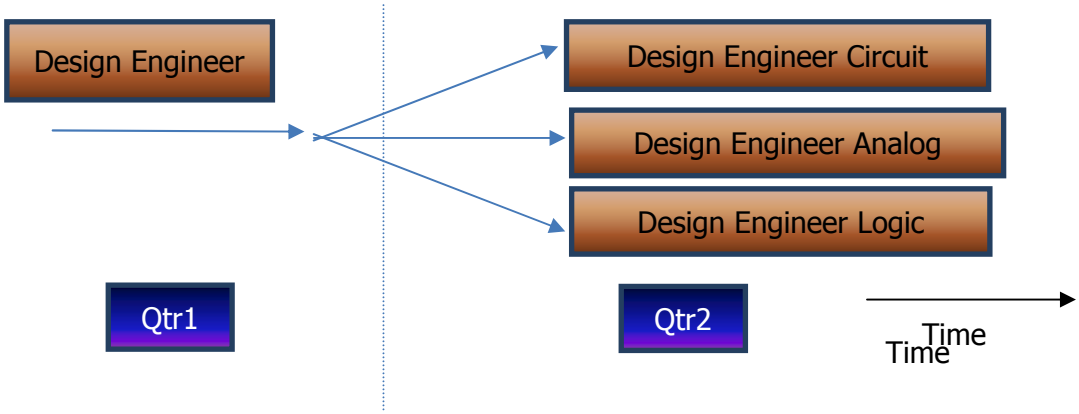


Figure 6: An illustration of the effect of Reorganization on data quality:

Case 1: splitting of skillsets

Figure 6 shows that the resources who used to get reported under skillset "Design Engineer" in quarter 1 and started getting reported under more granular skillsets in quarter 2.

Case 2: Merging of Skillsets: Contrary to the above example, multiple skills can be merged into a single super-skill category from a certain point of time.

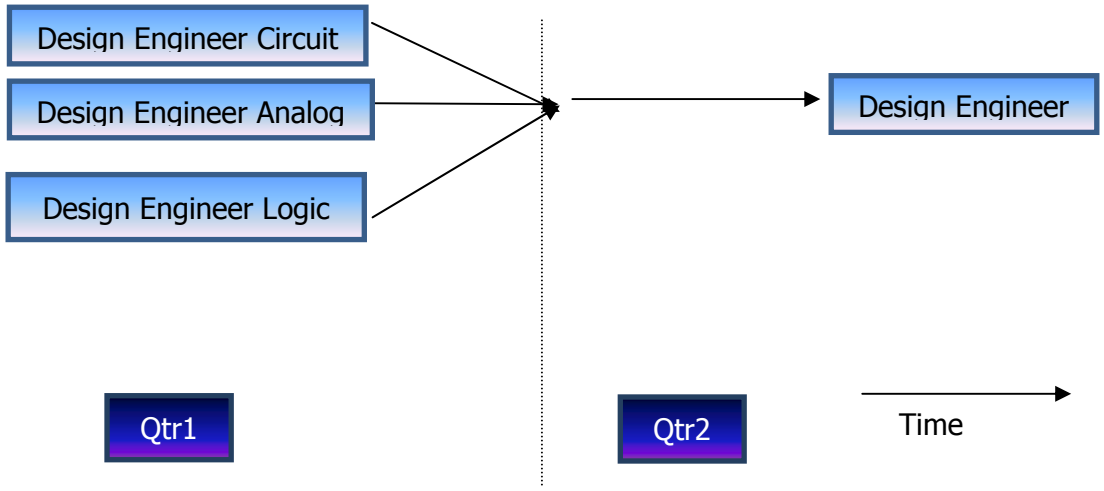


Figure 7: An illustration of the effect of Reorganization on data quality:
Case 2: Merging of skillsets

Case 3: Partial Split/Merge (More complicated):

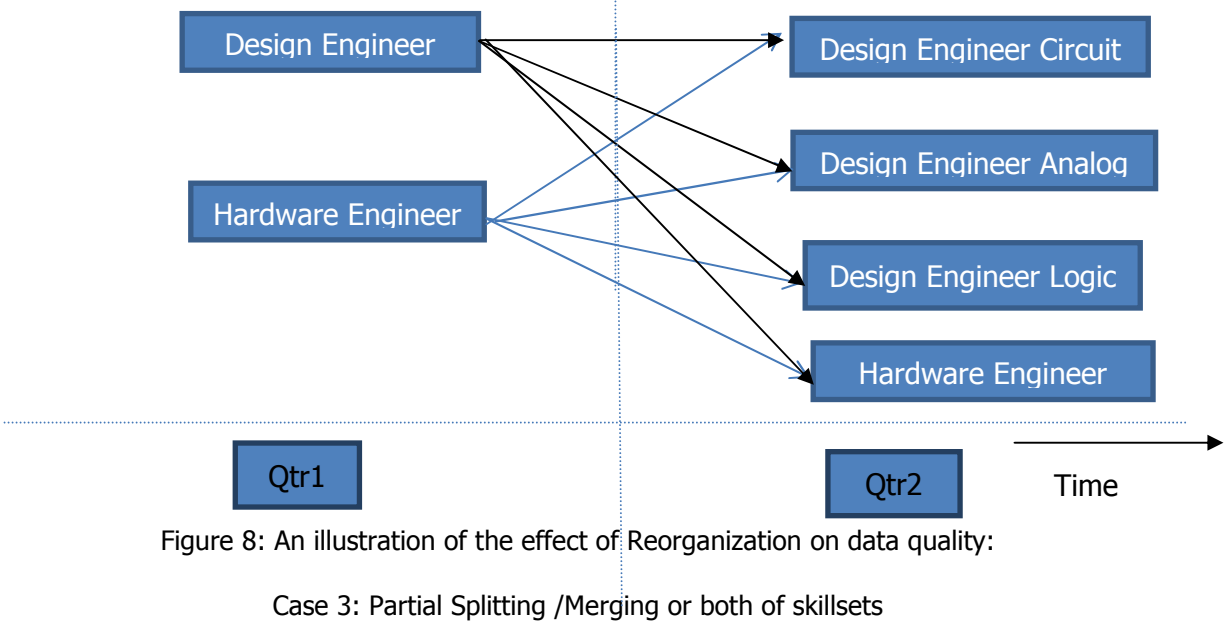


Figure 8: An illustration of the effect of Reorganization on data quality:

Case 3: Partial Splitting /Merging or both of skillsets

Figure 8 shows the picture of a complicated scenario of reorganization where the former skills are split and partially merged to form a new skillset in the following quarters.

For all of the 3 cases, if a project spans over quarter 1 and 2, a difficult question is how many "design engineer analog" resource worked for that project in each quarter?

For case 3, a harder question is how many "hardware engineer" worked in this project? This is even more difficult because although the name of the skillset is same, their scope/definition is changed over the quarters.

The hardest scenario is that this reorganization can affect different divisions/departments in different quarters. So there are possibilities that in a single quarter we will find skillsets with the same name but of different scope in different departments for the same project. (A project involves multiple departments responsible for different functional areas).

Solution:

The solution for this problem is to track the changes in skillsets at an individual level. The rationale behind this approach is that it is assumed that whatever the name of the skillset is over different quarters, a resource who worked as a "Design Engineering circuit" will continue or have continued to work as a "circuit design engineer" always in the company. Applying this logic 90% of the data was successfully interpreted. The worldwide employee Identification number (ID) is used as the unique identifier of the individuals to track the changes in their skill, division, project allocation.

Algorithm:

- ***Step 1:*** *For each employee ID track the current job role or the job role that the employee was assigned to right after the reorganization*
- ***Step 2:*** *Skip general managers, top level managers, architects who are covered by multiple projects with a small percentage of effort.*
- ***Step 3:*** *If there is no promotion (no transformation from designer/developer to manager) then go to step 4.*

  Otherwise go to Exceptions a) Promotions.
- ***Step 4:*** *Update earlier job roles with current job role/job role assigned right after the reorganization.*

Exceptions:

a) **Promotions:** There are exceptional cases like promotions although it is very less in number. If an employee gets promoted from "circuit engineer" to "Front end design manager" then the above rule won't be applicable. Those cases are handled separately. For those cases a minor modification is done to the earlier algorithm.

- ***Step 1:*** *For each promotion case, collect the skill the person used to get reported just before promotion say "skill A".*
- ***Step 2:*** *Find out what is the current skill of majority of people who used to get reported under "skill A" in that quarter(s) say "skill B"*
- ***Step 3:*** *Update that resources' earlier skills with "skill B"*

b) **Attrition:** Another type of exception case is the employees who left their job before the reorganization and have no continuation in timesheet entries over quarters after reorganization. We could not use those data for future use in forecasting.

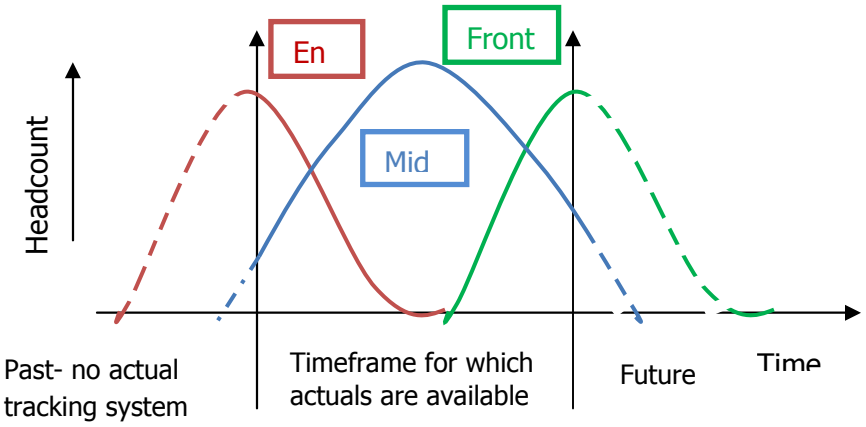
Similar type of rearrangements is seen in division as well as projects and program level. Those are solved by applying the above approach only.

Actual problems encountered at Intel become even more challenging when a reorganization has rearrangements and scope changes in skillsets, projects and divisions in all 3 dimensions. In each dimension the above approach is applied to minimize the effect of non-standardization in nomenclature.

3.2.5 Incompleteness of data due to long span of projects

Figure 9 describes this scenario. The actuals reporting system was started in 2009, so projects which started before that time have only their tail parts in the actuals. For the ongoing projects, the starting data is available but not the future part. There is another set of projects which have their middle part of the execution data available in actual tracking system, whereas their front parts executed before the actuals tracking system introduced and tail parts are to be

executed in future. As a result of this, as shown in the figure 9(a) none of the projects has a complete cycle captured in actuals system.



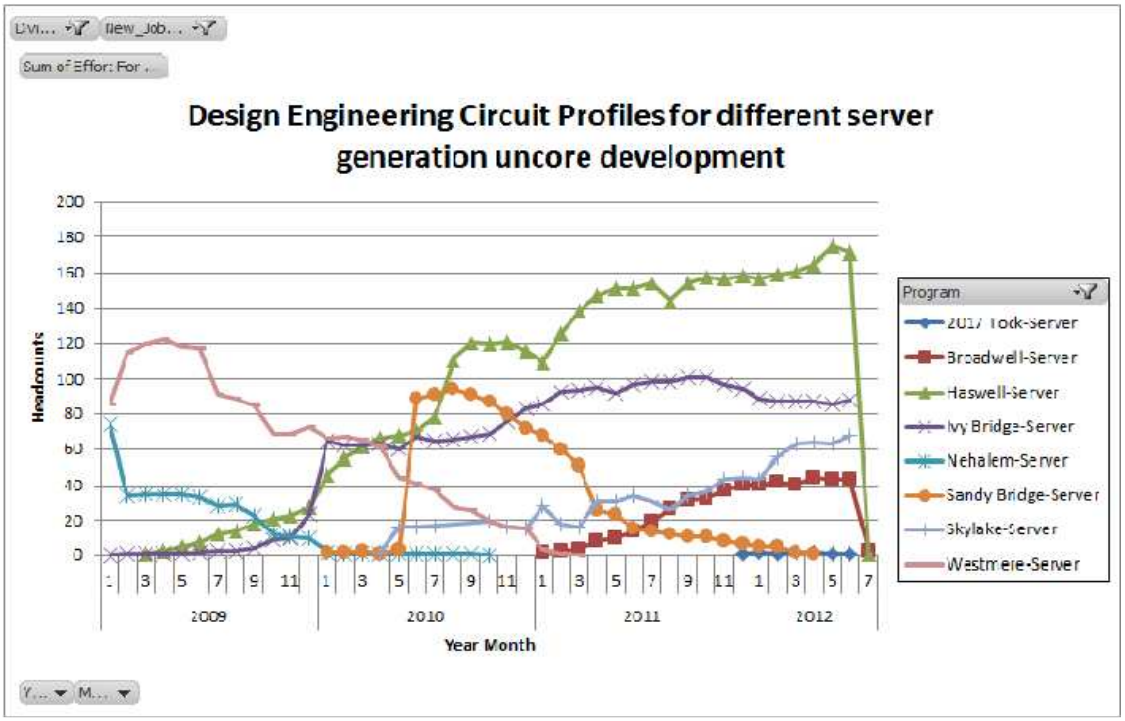


Figure 9 (a) and (b): Illustrates Data incompleteness due to longer span of projects than time period for which actuals are available.

Solution: One way to address this problem is to fill the data incompleteness by extrapolating the data. Though it will incur some error, it will be more useful than having nothing for that duration.

Extrapolation using curve fitting with boundaries restricted (to ensure the fitted curve tallies with the actual/planned start/end dates of the project) is very useful in filling data gaps. But after many rounds of trial and errors it is observed that the projects must have the following data in percentage available in actuals for a better extrapolation.

- more than 40% (either from front or from back)
- more than 30% (from the middle including peak)

Algorithm:

- ***Step 1:*** *Align the actual project execution data (curves) in either common start or common mid-point.*
- ***Step 2:*** *Plot missing start or end point. (By gathering information about project start or end date from product development roadmap databases)*
- ***Step 3:*** *Identify the patterns of the available part. Apply step 3a if a lot of misreporting/ under reporting is present in actuals.*
    - ***Step 3a:*** Apply weighted curve fit by allowing more weights to the peak points than base points to have a better accuracy, reduce the influence of bad/incomplete data. (This is an alternative step of 4a.)
- ***Step 4a:*** *Apply different curve fitting policies. e.g. linear, quadratic, cubic, Gaussian, cubic-spline fit etc.* (It is not recommended to go for higher degree fits beyond 3rd degree for polynomial and Gaussian beyond 1st degree as higher order fits cause over-fitting)
- ***Step 5:*** *Accept the fit which meets some threshold RMSE (root mean squared error) and if multiple fit meet RMSE cut off select the fit with lowest order.* (Selecting simple fit is an way to avoid over-fitting.)

Threshold level of RMSE is generally kept at .9 and for most of the projects either Gauss fit or polynomial fit met this RMSE criterion. If the RMSE of both Gauss fit and polynomial is very low, say .6 or below (rare case) then cubic spline fit is selected.

The reason behind applying more weights on the peak points is it is observed that there are almost negligible errors when peaks are reported. One possible explanation is when a project runs in full swing and utilizes maximum heads the people become more conscientious and as a result there are less error occurs. To reduce the effect of the data incompleteness from other non-peak points in the curve, peaks are loaded with more weight. The weight vector is a quadratic function which varies from 1 to 2 and assigns the peak point with the maximum weight and other 4-5 points surrounding peak with some weight more than 1 but less than 2.

Applying the above method the data incompleteness/gaps can be minimized. Figure 10 illustrates different curve fit methods applied on the actual data points to fill data gaps. Data Set 1 – 3 types of curve fits are applied – Polynomial – Cubic (C), Gaussian (G) and Cubic Spline (CS). Both Cubic and Gaussian failed to meet the RMSE criterion. So Cubic spline fit is used to fill the data gap from quarter 41 onwards. For data set 2 polynomial cubic fit worked well.

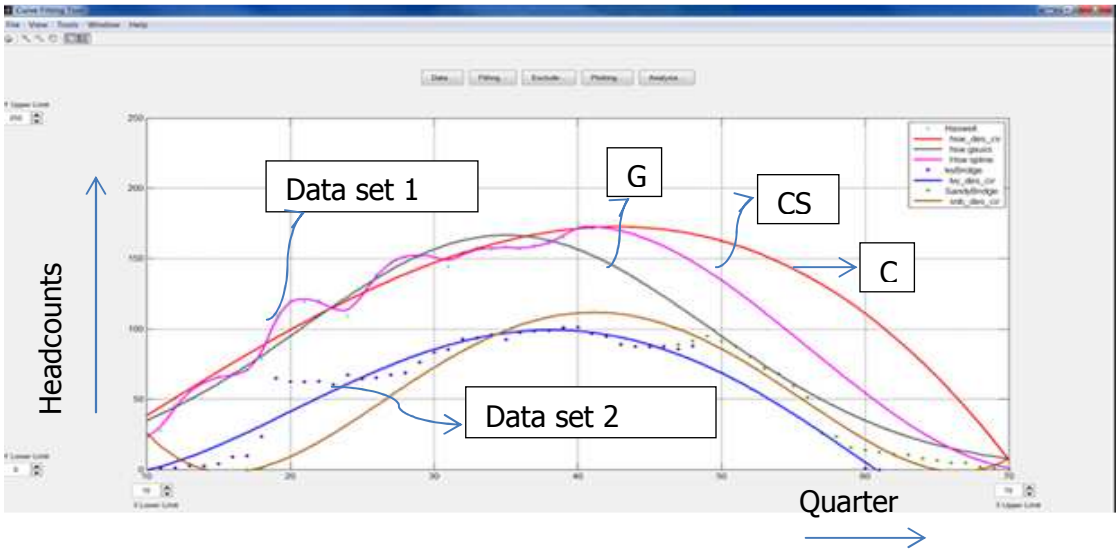


Figure 10: An illustration of the application of Extrapolation to eliminate data incompleteness and obtain complete project cycles

3.2.6 Determining the Proper level of aggregation of skillsets for better planning

Another challenge was to figure out the proper level of aggregation of the data which is useful for mining. There were 400 distinct skillsets before resolving reorganization and non-standardization in skillset nomenclature. After solving the nomenclature problem skills were reduced to 150. However 150 different skills are also too difficult to be used for planning and comparisons.

One possible approach is to use total heads of all skillsets for future prediction. But that is too abstract to forecast the resource requirement in design or in testing. So the next step is to determine the main functional areas of a project in the Semiconductor domain. It mainly involves two different types of works—Hardware and Software. Software and firmware work starts once the hardware work is 60% completed. There are different types of hardware work—development of product architecture and design (termed as front end work), platform work (integration of different chips into a platform), manufacturing work (termed as back end work) starts once the design work is almost over and validation work which also starts once significant progress is made in design and platform work. There were other skills like General Managers, Finance etc. For simplicity those resources were ignored because they are very small in numbers and that will increase the number of skillset super groups unnecessarily. So to have a better control on the data skill buckets are kept restricted to 5 only. Table 1 shows a few examples of the bucketing of detailed skills to 5 high level role categories to make the prediction model easy to use.

No.	Job Roles/Skillsets	Buckets/super group
1	Design Engineering Circuit	HW Front End Design and architecture
2	Component Architecture	HW Front End Design and architecture
3	Hardware(HW) Architecture	HW Front End Design and architecture
4	Board Repair and Rework	HW Back End Manufacturing
5	Product Development Engineering	HW Back End Manufacturing
6	Manufacturing Lab Technician	HW Back End Manufacturing
7	Board Design	HW Platform Engineering
8	CAD	HW Platform Engineering
9	Electrical Validation	HW Validation Engineering
10	System Validation	HW Validation Engineering
11	Compatibility Validation	HW Validation Engineering
12	Media	Software/Firmware
13	Graphics	Software/Firmware
14	BIOS	Software/Firmware

Table 1: Bucketing of detailed job roles into 5 super groups: some examples

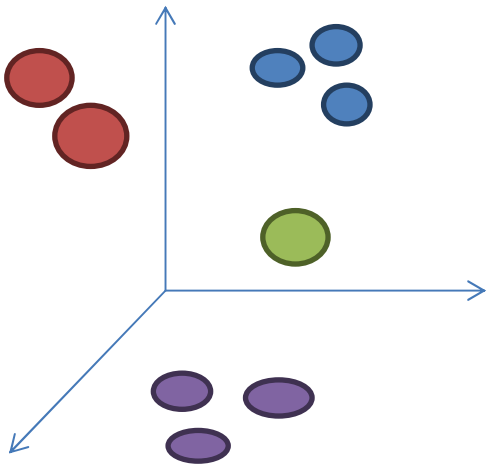
Chapter 4

DATA CLUSTERING

The question remains whether projects which are technically similar (belonging to same product family) show similar profiles for resource utilization. One method to decide this is to apply clustering on resource utilization data for different projects without giving any information about their product categories/ groups to the system.

Clustering essentially means grouping data which are similar from some perspective. In this thesis the perspective is resource utilization in different skills. The clustering dimensions (as described in the previous section 3.1.6) are

- Front end design engineers,
- Back end manufacturing engineers,
- Platform engineers,
- Validation engineers and
- Software/Firmware engineers.



Clustering Dimensions:
Project (Total Front end design engr,
Total Back end manufacturing engr,
Total Platform engineers,
Total Validation engineers,
Total Software/Firmware engineers)

Figure 11: Identification of similar projects in terms of skill-based resource utilization

4.1 Selection of Appropriate Clustering Algorithm

Two popular methods of clustering are K-means clustering and hierarchical clustering, each having its own pros and cons listed below.

K-means Clustering	Hierarchical Clustering (HAC)
Runtime: faster. Time complexity varies linearly with number of data points. O(n)	Runtime: slower. Time complexity varies polynomially (2^{nd} order) with number of data points $O(n^2)$
Cluster output and quality are highly dependent on initial seed selection.	No such case arises as it compares distances between every data points.
Iterative refinements are possible. I.e. a data point which is a member of one cluster in iteration 1 can be a member of a different cluster in iteration 2.	Iterative refinement is not possible; data points once assigned to a cluster can't change their clusters.

Table 2: Differences between K-means Clustering and Hierarchical Clustering

4.1.1 The Hybrid Approach: Algorithm

1. *Use 50% of the data points for HAC clustering and generate K clusters.* (Generally only 10% of the data is used for HAC but for this case as data points are less in number 50% of the data is used.)
2. *Take the medoids of those K clusters as the initial seeds for K-means clustering*
3. *Run K means clustering on 100% of the data with those initial seeds generated from applying HAC on a reduced set of data.*

The advantage of the hybrid approach is to increase the probability of a good set of seed selection for K-means. On one hand, the goodness of K means clusters are highly sensitive to initial seeds. On the other hand, the disadvantage of HAC is it is computationally expensive on the full dataset. That is why HAC is used on a small dataset to identify the initial seeds for K

means. Using K means on the full dataset with a good set of initial seeds ensures better performance, goodness of the clusters and iterative refinement of the clusters.

4.2 Cluster selection strategy

The Centroid distance method is used as a cluster selection strategy. It is obvious that the centroid of a cluster will bear the characteristics of the cluster (here a cluster means a group of projects) and K means generally generates globular clusters. So the centroid method is more applicable for this problem than other methods like maximum distance, minimum distance, ward distance, etc.

4.3 Distance Measure

Among different distance measures like Euclidian, Cosine, Manhattan distance, Euclidian distance is chosen. If Euclidian distance is selected the impact of variance will not be ignored. What is the impact of variance in the given data?

Dimensions	Variance (total)	Variance (quarterly max)
Front End Heads	6572	508
Back End Heads	1378	167
Platform Heads	780	142
Validation Heads	1463	147
SW/FW heads	1070	112

Table 3: The variances in different dimensions of clustering

From Table 3 it is clear that the dimensions chosen have wide difference in their variances. The variance of front end design heads especially is very high relative to other dimensions. The reason behind selecting Euclidian distance is to give equal importance to all dimensions as they are all headcounts only of different skills. For successful project completion availability of heads from all skills are equally important. So to reduce the importance of the dimension having high variance will go against business requirement.

4.4 Selection of Perspective of clustering

There are a number of ways in which the projects can be compared and clustered.

- Project similarity in terms of total work-done i.e. **total headcounts** required in each dimension or skill bucket for a complete project cycle:
- Project similarity in terms of maximum work-done i.e. **peak headcounts** required in each dimension or skill bucket for a complete project cycle:
- Project similarity in terms of duration i.e. **counts of quarters** required in each dimension/phase for a complete project cycle:
- Project similarity in terms of percentage distribution of different skills in total heads i.e. **(number of heads in different dimension / total headcount in all skills)*100** required in each dimension or phase for a complete project cycle:

Total headcounts by skillset comparison is most useful to compare the volume of the work and it is observed generally that projects with high total headcounts have high peaks and long duration. However, peak headcount and duration comparison is important to identify the odd cases as two projects having same total work can vary in peak and duration.

4.5 Clustering Output, Observation and analysis

4.5.1 Results of Clustering on total headcounts in different skillsets: Data points are grouped into 6 clusters. Number of iterations: 2. Within cluster sum of squared errors: 1.287. Results in detail are shown in table 4.

Project	Family	BE_HC_TOT	FE_HC_TOT	Plat_HC_TOT	Val_HC_TOT	SW_HC_TOT	Cluster
Project A	Family A	311	1481	69	1127	246	0
Project J	Family G (B)	1423	2071	136	1091	116.4	0
Project P	Family H (C)	610	2531	295.4	1091	260	0
Project B	Family A	97	563.6	27	443	16	1
Project C	Family B	50	1408	5	306	371	1
Project F	Family D	116	1165	70	233	12	1
Project I	Family F	191	728	91	305	183	1
Project K	Family G (A)	0	3	8	23	10	1
Project Q	Family H (A)	0	7.6	104	20.7	43.3	1
Project R	Family H (A)	0	9	105.5	88.8	38.3	1
Project E	Family C	664	1586	82	443	125	2
Project N	Family H (C)	586.8	1242.3	44.4	494.4	10.9	2
Project O	Family H (C)	542	827	65	348	1.3	2
Project L	Family H (B)	1378.5	5051.9	738	1483.2	278.8	3
Project D	Family C	1179	6575	88	571	136	4
Project G	Family E	1329	3586	288	1002	1071	4
Project H	Family E	1329	4407	255	1058	735	4
Project M	Family H (B)	1265.4	3308	786.4	1015.4	360	5

Table 4: Result of clustering projects on total headcounts in each skillset for entire project cycle

Observation: Projects of the same category don't follow the same resource utilization pattern. Rows highlighted in the table 4 shows projects of same family going in different clusters as explained below.

- Though "Project L" and "Project M" are both from "Family H (B)" they went into different cluster.
- Though "Project P" was a "Family H (C)" it went to cluster 0 while other "Family H (C)" projects are in cluster 2. So "Project P" is more similar to "Family G (B)" resource profile.
- Two member of "Family C" group went into different clusters 4 and 2.
- Two "Family A" projects went into different clusters 0 and 1. One "Family A" project (member of cluster 0) resembles a "Family G (B)" server project profile.
- Some of the projects of same family maintained consistency in being in the same cluster. E.g. Project Q and Project R (family H (A)) both are members of cluster 1. Project G and Project H (family E) both are members of cluster 4.

Inference: Therefore, the results are confirming that a single resource utilization template to describe each product family is not possible for all families.

4.5.2 Results of clustering data on max headcounts

The output of the clustering is 6 clusters with a within cluster sum of squared error of 1.153. The cluster assignments that took 4 iterations to converge are shown in Table 5 in detail.

Project	Family	BE_HC_MAX	FE_HC_MAX	Plat_HC_MAX	Val_HC_MAX	SW_HC_MAX		Cluster Nu
Project J	Family G (B)	166.8	207	13.6	123	20		0
Project P	Family H (C)	71.4	256.5	34	123.8	29.4		0
Project D	Family C	134	593	6.8	70	15.6		0
Project Q	Family H (A)	0	1.2	18.5	4.6	9.8		1
Project R	Family H (A)	0	2	18.2	17	8		1
Project B	Family A	13.6	67	3.8	60.2	3		1
Project K	Family G (A)	0	0.7	1.8	4.6	3.4		1
Project F	Family D	14.2	110	6.8	29.1	2		1
Project C	Family B	3.9	185.1	1.1	31.5	40.4		1
Project N	Family H (C)	55.4	113.2	3.8	49.1	0.9		2
Project O	Family H (C)	62.8	67	25	48.8	0.5		2
Project A	Family A	28.4	172.5	7.3	93.4	31		2
Project E	Family C	73.5	162	7.5	59	14.2		2
Project I	Family F	27	134.8	17	56	26		2
Project L	Family H (B)	129.2	466	97.5	147.3	37.5		3
Project G	Family E	118.1	508.7	35.6	124.4	112.3		4
Project H	Family E	118.1	407.9	35.6	114.2	91.6		4
Project M	Family H (B)	146	269	101.5	146	37.3		5

Table 5: Result of clustering projects on maximum headcounts in each skillset for entire project cycle

Observation: Most of the projects remain in same cluster when their skill-based maximums of headcounts per quarter for entire project cycle are considered; few of them changed their places. "Project A" (of "Family A") earlier was with "Family G (B)" projects but now shifted to "Family H (C)" projects. So though in total work volume it is similar to "Family G (B)", in terms of max heads it is similar to "Family H (C)". Still two "Family A" projects are placed into different clusters. In a similar way, one "Family C" project is moved into "Family G (B)"cluster while in the total head scenario it was with "Family E".

Inference: Therefore, the results are confirming that a single resource utilization template to describe each product family is not possible for all families.

4.5.3 Results of clustering data on project durations:

Output: Number of clusters are 6. It took 5 iterations to converge. Within cluster sum of squared errors is 1.211 which indicates the goodness of clusters. The results in detail are shown in Table 6.

Project	Family	BE_HC_DUR	FE_HC_DUR	Plat_HC_DUR	Val_HC_DUR	SW_HC_DUR		Cluster Nu
Project C	Family B	9	23	0	17	18		0
Project I	Family F	11	12	9	12	14		0
Project E	Family C	9	6	5	9	5		1
Project K	Family G (A)	0	2	2	4	2		1
Project Q	Family H (A)	0	0	9	4	7		1
Project R	Family H (A)	0	1	9	4	7		1
Project B	Family A	11	18	6	15	4		2
Project F	Family D	13	21	13	15	4		2
Project N	Family H (C)	19	22	12	19	0		2
Project O	Family H (C)	18	15	5	16	0		2
Project A	Family A	23	22	15	26	18		3
Project D	Family C	22	26	20	18	16		3
Project G	Family E	24	20	15	15	20		3
Project L	Family H (B)	22	22	17	21	18		3
Project P	Family H (C)	17	23	16	19	17		3
Project H	Family E	24	19	12	20	16		4
Project J	Family G (B)	22	22	16	19	10		5
Project M	Family H (B)	20	17	18	17	16		5

Table 6: Result of clustering projects on duration for each skillset related work for entire project cycle

Observation: One important observation is when projects are clustered in terms of duration of different skill-works, most projects showed similar patterns of having duration of more or less 4-5 years. Some of the Family H projects and their accessories Family A are part of same cluster whereas they widely differ in total and max head comparison.

While collecting duration data of the projects there is a need to set a cut-off resource utilization. Gaussian curves never go to zero and can go infinitely long with negligible value in both direction which will generate misleading information about project duration. To mitigate this issue a cut off is set up which is 5% of the max headcount.

4.5.4 Results of clustering of projects on percentage utilization of resources per skill:

Project	Family	BE_HC_%	FE_HC_%	Plat_HC_%	Val_HC_%	SW_HC_%	Cluster_nu
Project A	Family A	9.62	45.79	2.13	34.85	7.61	0
Project B	Family A	8.46	49.15	2.35	38.64	1.4	0
Project L	Family H (B)	15.44	56.57	8.26	16.61	3.12	0
Project M	Family H (B)	18.79	49.12	11.68	15.08	5.35	0
Project P	Family H (C)	12.74	52.87	6.17	22.79	5.43	0
Project K	Family G (A)	0	6.82	18.18	52.27	22.73	1
Project R	Family H (A)	0	3.73	43.68	36.76	15.83	1
Project E	Family C	22.9	54.69	2.83	15.28	4.31	2
Project J	Family G (B)	29.42	42.81	2.81	22.55	2.41	2
Project N	Family H (C)	24.67	52.22	1.87	20.78	0.46	2
Project O	Family H (C)	30.39	46.37	3.64	19.51	0.07	2
Project D	Family C	13.79	76.91	1.03	6.68	1.59	3
Project F	Family D	7.27	72.99	4.39	14.6	0.75	3
Project C	Family B	2.34	65.79	0.23	14.3	17.34	4
Project G	Family E	18.27	49.29	3.96	13.77	14.72	4
Project H	Family E	17.07	56.62	3.28	13.59	9.44	4
Project I	Family F	12.75	48.6	6.07	20.36	12.22	4
Project Q	Family H (A)	0	4.31	59.25	11.76	24.68	5

Table 7: Results of clustering of projects on percentage utilization of resources per skill

Observation: Here the majority of projects show consistency with other members of the product family in maintaining similar skill-based percent utilization. But some of them still did not cluster as expected e.g. "Family C", "Project P" and "Family H (A)".

Inference: As the majority showed good pattern here, it gives us an approach to predict the total headcount and apply the percent distributions to get different skill-based heads. But accuracy will be compromised as we can see within a cluster; there are wide differences in percentages (7%-20%). We have very few data points to conclude that % distribution remains constant within a product family.

4.5.5 Summary of Inferences of all clustering approaches

Inference: Therefore, it is clear from these 4 ways of clustering that projects belonging to the same product family can show widely different resource demands. To create resource requirement templates, there might be factors to consider other than their high level

category/family. The cluster outputs led to consulting with human experts (program leaders) to explain the reasons for those anomalies.

There is a possibility of another way of comparison – the ordering in which different functionalities of a project are performed. But the dataset used here mostly showed the same order e.g. early involvement of front end design folks, followed by platform work, followed by manufacturing and validation and software work. So no clustering was done using that perspective.

Chapter 5

FEATURE COLLECTION

So far we are able to do a coarse level segregation of projects using their product family. After doing clustering from different perspectives we concluded that using only product family as a feature is not sufficient to create resource requirement templates with sufficient accuracy. The important question is what additional factors make the technically similar projects different in terms of resource requirements? In other words what are the features that impact resource demands of different skills for a particular project?

5.1 Some background on Semiconductor products

In every industry, products change over time as the world changes, people change and their demand/expectation change. The Semiconductor industry is no exception. As time flows, to meet peoples' increasing expectations and handle market competition, chip manufacturers provide more features called RAS features (Reliability, Availability and Serviceability features) to every generation of product. While making chips with improved features like higher performance, lower power, more memory etc., it in turn increases the complexity of the product.

In Intel, there are two types of improvements that are done on server products, as detailed below:

- Tocks [Reference 17] – involve a major change in architectural level or introduction of new microarchitecture which results in better performance, higher speed, more memory, better I/O etc. Basically it is a change in circuits and features keeping the manufacturing process same. This requires larger engineering efforts.
- Ticks [Reference 17] – are basically a manufacturing process improvement keeping the circuits and features much the same. This provides performance improvement only in gaining speed. This is generally done by shrinking the Die Size with the same number of transistors in it. Ticks generally require smaller engineering efforts than tocks.

5.2 Features from Product Roadmaps / wiki

The technical features of the products which impact the complexity of the product and hence impact resource utilization are collected from product roadmaps or wiki, as given below:

- **Core Count –** Number of cores per die of a chip keeps increasing generation after generation. More cores mean more parallel processing and hence more execution speed for a microprocessor.
- **Die Size –** The size of the die is in mm^2. For tick products, shrink in a die size keeping the core count same, is an indicator of increasing complexity. While for tock products, expansion in die size is an indicator of increased complexity, as more components are added to the chip to provide more features, it in turn increases the die size. So change is Die Size alone cannot indicate product complexity.
- **Core Count Die Size Ratio (derived):** As none of the core count or die size can predict product complexity alone, a derived feature from die size and core count which is proportional to number of transistors on a die is used. Core count to die size ratio keeps on increasing over time (generation after generation) for every server product family. So this can act as a feature to indicate improvement of the product or increase in the complexity of the product. As this ratio goes higher the product and the project becomes more complicated.
- **Number of New Sockets –** Sockets provides the connection electrically and mechanically between a microprocessor chip and the platform (circuit board). The complexity does not change if there is a reuse of sockets from earlier generations. However, if there is any introduction of new sockets in the product then project complexity goes higher.
- **Die Package Combination –** The package is the substrate on which the die is placed and integrated with other components. If dies of different sizes are to be placed on the

same package as a part of product release, that increases significantly the platform work and validation work, with an increase in the project complexity.

- **Cache Memory Size –** Though higher cache memory indicates more memory and improvement in product, it is 100% correlated with core count. So in regression the use of core count as a feature is sufficient.

5.3 Project Experts' Feedback

According to human experts the above mentioned features are not enough to determine the increase in complexity of the product due to architectural changes (mainly for tock products). The features given by the human experts are listed below:

- **First Generation Memory Technology:** When there is an introduction of a new memory architecture e.g. DDR3 (double data rate type three synchronous dynamic random access memory) or DDR4 (double data rate (fourth generation) synchronous dynamic random-access memory) in the microprocessor it will greatly increase the complexity of the product. This in turn increases the resource requirement for that project especially the hardware front end design people.
- **First Generation PCI (Peripheral Component Interconnect) Technology:** A new I/O architecture also increases the number of design engineering people required for the project but the impact is less than that of a new memory technology.
- **First Generation "other" Technology:** Apart from memory and I/O, there can be implementation of new software technologies, low power technologies, etc. This also increases complexity of the project and design engineer requirement but the impact is lower than memory and I/O changes.
- **Introduction of new core technology:** Generally Client precedes Server products and the core developed during the client chip manufacturing is reused in server. But if in any case this does not happen then it increases the amount of design heads significantly in server projects.

- **Un-core (server chip components other than core) Reuse:** Apparently, next generation products should take more heads/resource than its earlier generation as the next generation product is more complex and contains more features. But some exceptions were found while doing clustering. The reason behind taking less heads for a future product with additional/improved features is **Reuse.** There can be different ways of reuse either in concept, architecture, logic, code component, hardware blocks, microchips etc. Depending on the level of reuse heads from various skills are reduced. For example,
    - Architecture, concept reuse reduces hardware architects,
    - Design reduce reduces front end design engineers,
    - Building block reuse reduces design as well as manufacturing heads,
    - Software firmware code reuse reduces software heads.

    As it is obvious that core is always reused in servers from corresponding clients, reuse of components comes in the un-core part of server projects (anything in the die other than core is called un-core). The feature stores the reuse value as a percentage reuse of total un-core work.

Chapter 6

MODEL BUILDING, ITS VALIDATION AND SELECTION

There are a number of data mining learning methods available to predict a target based on input features including Support Vector machines (SVM), Regression, Artificial Neural Network (ANN), Decision trees, K nearest neighbor, Naïve Bayes classifier, etc. Some of them are effective for predicting discrete targets (e.g. binary variables, categorical variables, class variables etc.) and some are effective in predicting targets which are continuous in nature. Among those learners linear regression is selected to be used to build the model. The main reason behind selecting regression is the output variable is continuous. There is a challenge for this model since there are not many data points for training. So in that respect linear regression is more intuitive and controllable than other data mining methods like SVM or ANN.

Figure 12 illustrates the high level representation of the regression model. In this figure the projects are represented in two different spaces. On the left hand side the projects are represented in terms of their technical features (as described in Chapter 5) and on the right hand side the projects are represented in terms of their resource utilization in different skills (as described in section 3.1.6 and chapter 4). There will be a transformation function which will map projects (data points) from the left space to resources from the right space and developing that function is the focus of this chapter.

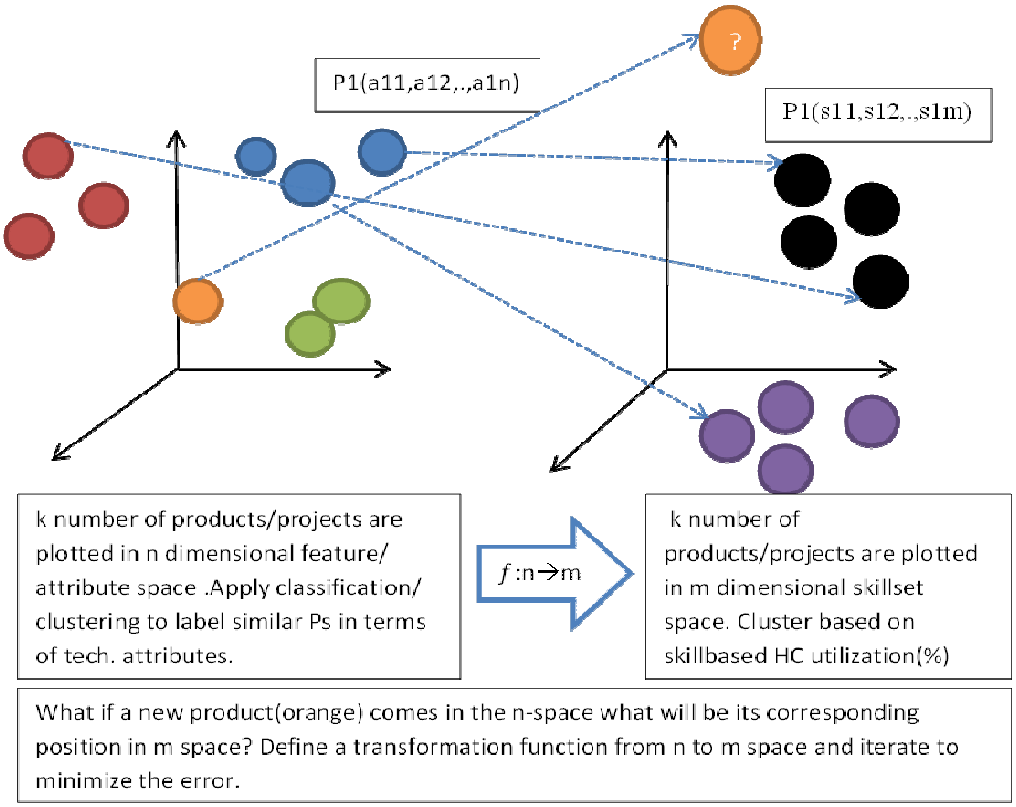


Figure 12: Illustrates the high level diagram of the model, the plotting of projects in technical space (left) and resource space (right) and their mapping

6.1 Factors considered in Model building approach:

Some of the factors considered during model building are enumerated below:

1) **Concept of complexity Score:** In this thesis there are 5 different skillsets for which the resource demand has to be predicted. This indicates the requirement of building a model having 5 target variables. As there is a small number of project data available for training the model, a simple model with one output variable is preferred over a multi-variable model. The approach used here is to predict a complexity score for each project and then derive

headcounts per skills from those scores (plotting skill based headcounts against scores, using curve fitting explained in section 6.5).

2) **Linear relationship between Complexity Score and front end design heads:** A linear relationship between front end design headcounts and complexity scores is developed. It is observed that as projects become more complicated generation after generation, their front end design head populations vary widely and that skill demand is most difficult to predict, while resource requirements of other skillsets are more constant across projects (please refer Table 3 to have the variance in resources in each skillset). The advantage in doing this is to reduce the variance of the target variable (front end head varies from 10 to7000) .Enough training samples with a good distribution over the target intervals are not available. That is why complexity score is used as target variable which varies only from 0-150 which is comparable with the variance of input variables as well.

3) **Transformation of categorical input variables:** In this thesis, input variables only have one categorical variable -- "product family" which contains values like "Family A", "Family B", etc. Linear regression is not good in dealing with categorical input variables unless and until any indicator function is used to convert them into numeric. By using indicator function some numeric values were generated against those categories which indicate category A is somewhat different from category B. However, how much different the categories are from each other, was missing in the numeric values. Therefore, to have a better rational transformation a numeric variable "product family" is introduced which contains the inherent difficulty level of that category in terms of resource requirement. Table 8 shows the transformation of categorical variable into an interval variable. This indicates the inherent difficulty level in designing and manufacturing any product belonging to a family. These scores were assigned by the program experts and higher score indicates more difficulty. In this way a numeric distinguishing feature is generated which distinguishes the target at a coarse level.

Family category renamed	family
Family D	30
Family A	40
Family G (B)	50
Family E	70
Family H (C)	80
Family H (B)	80
Family C	90

Table8: Derivation of interval variable family from categorical family_category

6.2 Data used in training the model

Table 9 shows the data which includes the project name with their project family and other technical features. The right most column represents the front end design complexity of the project which is the predictable target (FE_COMPLEXITY).

ID	Project Name	Product Family category	Project Family	Core Count	DieSize (mm^2)	CC Die ratio	Uncore Reuse (%)	FE COMPLEXITY (y)
4	Project B	Family A	40	1	59	0.017	85	12
5	Project I	Family F	45	8	104	0.077	10	16
6	Project O	Family H (C)	80	8	416	0.019	70	18
7	Project N	Family H (C)	80	18	664	0.027	70	26
8	Project A	Family A	40	1	128	0.008	50	31
9	Project E	Family C	90	18	664	0.027	70	33
10	Project J	Family G (B)	50	12	543	0.022	0	42
11	Project P	Family H (C)	80	15	543	0.028	40	52
12	Project M	Family H (B)	80	8	416	0.019	0	67
13	Project G	Family E	70	62	729	0.085	30	73
14	Project H	Family E	70	72	726	0.099	20	89
15	Project L	Family H (B)	80	18	664	0.027	0	102
16	Project D	Family C	90	8	557	0.014	0	133
17	Project F	Family D	30	1	35	0.029	0	25
3	Project Q	Family H (A)	80	4	177	0.023	95	1
2	Project R	Family H (A)	80	4	283	0.014	95	1
1	Project K	Family G (A)	50	4	160	0.025	97	1

ID	Project Name	New_core	New tech sw power other	Die Package combo	First gen memory tech	First gen PCI	Num new sockets	FE_COMPLEXITY (y)
4	Project B	0	0	0	0	0	0	12
5	Project I	0	1	0	0	0	0	16
6	Project O	0	0	1	0	0	0	18
7	Project N	0	0	1	0	0	0	26
8	Project A	0	1	0	0	0	0	31
9	Project E	0	0	1	0	0	0	33
10	Project J	0	0.5	6	0	0	0	42
11	Project P	0	1	1	0	0	1	52
12	Project M	0	1	5	0	1	1	67
13	Project G	0	1	1	0	0	0	73
14	Project H	0	1	1	0	0	0	89
15	Project L	0	1	6	1	0	0	102
16	Project D	1	1	1	0	0	1	133
17	Project F	0	1	0	0	0	0	25
3	Project Q	0	0	0	0	0	0	1
2	Project R	0	0	0	0	0	0	1
1	Project K	0	0	0	0	0	0	1

Table 9 (a) and (b): Project Data with their Technical features

6.3 Feature Selection:

The objective of feature selection is to pick the best set of features which are effective in distinguishing and predicting the target variable. This can be done in three ways: i) Unsupervised, ii) Supervised and iii) Semi Supervised. This thesis dealt with only unsupervised and semi-supervised features selection.

i) **Unsupervised Selection:** Unsupervised approach is based only on statistical analysis of the features which does not involve any human intervention or domain experts' knowledge. The statistical method used for unsupervised selection is stepwise selection. The algorithm is explained below.

Stepwise selection [Reference 16]**:** In stepwise regression the system keeps adding features iteratively or in steps. Initially it starts with no regressor but with just the intercept (constant). Then it keeps adding one feature at a time giving priority in selecting features which are highly correlated with the target/response variable. Every time it picks the most correlated feature with the adjusted target (the residual target obtained after adjustment of earlier features). If that meets the significance threshold level then that feature gets added into the regressor subset.

At the same time it calculates the significance level of each pre-selected features. In further iterations if any of the pre-selected features doesn't meet the significance threshold level the system removes it. That is how stepwise selection works forward and backward and comes up with the optimum regressor subset.

Parameters:

1) Selection based on features with least p values (i.e. more significant in distinguishing target). The significance threshold used is .06 for entry and .06 for staying.
2) Elimination of highly correlated features among themselves where their correlation is more than .75 in both ways (positive or negative).

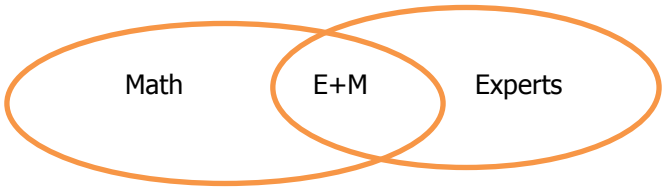
Results: Table 10 shows the list of features selected stepwise.

Step	→→→	1	2	3	4	5	6	7	8
constant		73.12	13.20	19.54	19.64	22.31	27.79	27.11	40.38
Un-core reuse	Coeff	-0.712	-0.761	-0.662	-0.542	-0.414	-0.350	-0.255	-0.340
	P-Value	0.002	0.000	0.000	0.001	0.001	0.003	0.003	0.000
family	Coeff		0.93	0.73	0.49	0.30			
	P-Value		0.005	0.017	0.051	0.133			
New_Core	Coeff			48	64	78	90	71.6	67.4
	P-Value			0.051	0.005	0.000	0.000	0.000	0.000
Core Count	Coeff				0.62	0.72	0.82	0.90	1.32
	P-Value				0.014	0.001	0.000	0.000	0.000
First gen memory tech	Coeff					43	51	58.7	49.5
	P-Value					0.009	0.003	0.000	0.000
Number of new sockets	Coeff							27.1	20.8
	P-Value							0.002	0.001
CC_die ratio	Coeff								-430
	P-Value								.002
S		28.1	21.8	19.4	15.5	11.7	12.5	8.37	5.42
R-Sq		49.92	72.05	79.37	87.83	93.61	92.08	96.75	98.76
R-Sq(adj)		46.58	68.06	74.61	83.11	90.70	89.44	95.28	98.02
Mallows Cp		315.5	172.3	126.3	72.8	36.9	45.0	16.3	5.1

Table 10: List of features selected by Regression (stepwise)

Feature Conflict between Math and Experts:

Some of the features selected by the tool are refuted by the program experts as they don't consider those features at all while planning engineering resources. e.g. core count.



Feature	Datatype	Type	Selected by Math & human(YY/YN/NY/NN)	Sign of coeff. conflict
CC_Die_ratio	interval	input	**YY**	**yes**
Uncore_Reuse	interval	input	**YY**	no
New_core	binary	input	**YY**	no
first_gen_memory_tech	binary	input	**YY**	no
Num_new_sockets	binary	input	**YY**	no
CoreCount	interval	input	**YN**	
family	interval	input	**NY**	
DieSize	interval	input	**NY**	
New_tech_sw_power_other	binary	input	**NY**	
Die_Package_combo	interval	input	**NY**	
first_gen_PCI	binary	input	**NY**	
FE_complexity	interval	target		

Table 11: Shows the conflicts of features selected by Math and human experts

Table 11 shows there are some features which both Math and human experts consider important in resource plan (marked by YY). Some features are selected by math but refuted by human experts (YN). Some are considered important by humans but rejected by Math (NY).The rests are rejected by both (NN).

For those cases in which both accepted as important there can be a conflict in sign of the coefficient. Sometimes when human intuition/business knowledge says feature and target are positively correlated, Math says negative correlation (Example CC_Die_Ratio).

Forceful use of some features given by experts:

As per human experts' feedback, some of the features are forced into the model making their use mandatory. Those are marked with priority level 1, 2, 3 in Table 12. The rest of the features are passed as free features allowing system to select the best subset according to their significance level.

Feature	Datatype	Type	Priority
Family*Uncore_Reuse	interval	input	Must(1)
Family*Die_Package_combo	interval	input	Must(2)
CC_Die_ratio	interval	input	Must(3)
New_core	binary	input	Free
New_tech_sw_power_other	binary	input	Free
Family	interval	input	Free
first_gen_memory_tech	binary	input	Free
first_gen_PCI	binary	input	Free
FE_complexity	interval	target	Must

Table 12: Final list of features which are passed into the system for model building

The technical reason behind using a combination of features or use of product of features as single features (e.g. Family*Die_package_Combo) is mainly to increase the significance level of the input variable. The features which shows higher P values alone, if combined with a lower P value feature the significance level of the derived feature improves (the P values against each feature is given in table 10).

Real world interpretation of those combined variables:

Family*Uncore_Reuse = The feature Uncore_reuse stores the reuse in % which alone does not indicate how much of total actual work was made easier by this reuse for variety of product. Within a family we can say the effect of reuse is uniform, as reuse goes up product complexity goes down in same ratio. But across multiple product family (starting from small chips to large chips) the effect of reuse is not the same in magnitude. So the combination of Family and reuse i.e. Family*reuse indicates efficiently how much the actual work was made easier by the reuse in different product categories.

Family*die_package_combo = Rationale behind using this feature is similar to the reason described above. There is an inherent difference in difficulty between different product families. Die_package_combo indicates the variety of technical specification available for a particular

product (e.g. Product L has 5 different variety due to die size variation). So the combined feature including two will be more appropriate for predicting the target.

The features thus obtained are passed to the regression tool to build the model and train the model.

6.4 Model Building:

Multiple models were built using the Linear Regression method to predict one target (i.e. the complexity of the project) and the best possible one is selected for future use:

6.4.1 Attempt 1: Building the Model with an Unsupervised approach for feature selection: (Whatever Math finds the best)

In table 10, the features selected by a stepwise Regression process as a best subset of distinguishing features to predict the target are listed. The equation given below provides a linear relation between target and input features. The product of the corresponding coefficient and feature indicates the magnitude of the feature's impact on the target and the sign indicates the impact is in a positive or negative direction. In the equation given below the features in bold are explained further in terms of their effect on the target in sign and magnitude.

FE_COMPLEXITY (y) = 40.4
+ **1.32 * CoreCount**
- 0.340* Uncore_Reuse
+ 67.4* New_core
- 430 * CC_Die_ratio
+ 49.5 * first_gen_memory_tech
+ 20.8 * Num_new_sockets

Goodness of fit : S = 5.41531 R-Sq = 98.8% R-Sq(adj) = 98.0%

Though this model produced good R^2 and $R^2_{adjusted}$, human experts didn't accept the model. The reasons include:

- Use of feature CoreCount which human experts claim they never consider while planning resources.

- The use of a negative coefficient against "Core Count Die Size Ratio" goes against human intuition. Intuitively with the increase of that ratio the product becomes more complex which incurs more resources, especially front end design engineers.

The other features meet the human expectation in terms of their impact on the target in both magnitude and direction. For example,

- the introduction of a new memory technology increases the complexity by 49.5
- the introduction of a new core increase the target by 67.4
- each new socket (if not reused) increases the complexity by 20.8
- reuse decreases the project overall complexity by .34 times the percentage reuse.

However as explained earlier, experts were unhappy because the factors which they consider as important while resource planning weren't picked by the math.

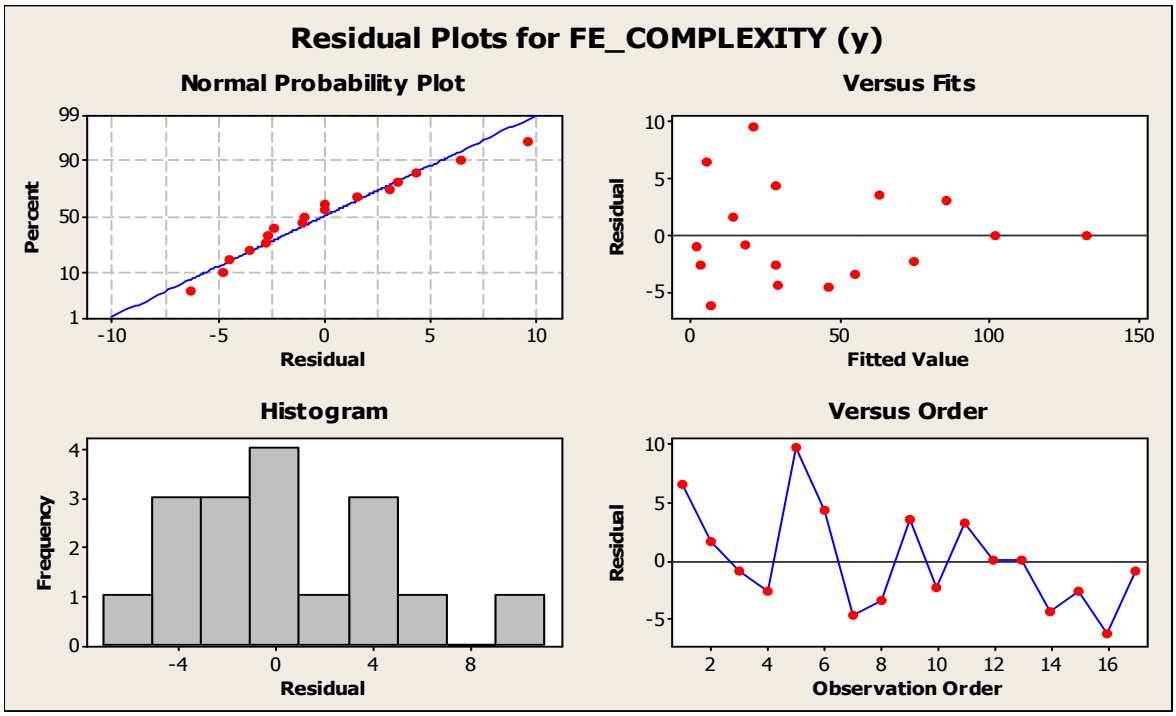


Figure 13: Illustrates the residual plots for model built using unsupervised feature selection

In Figure 13, the normal probability plot and inward funnel shaped residual vs. fitted value plot indicate there is a non-linearity in the target data at its low end. As human experts

didn't approve it and there is also a non-linearity in a linear model, another attempt was made to build a better model.

6.4.2 Attempt 2: Semi-Supervised (mandatory use of some features given by experts)

In a semi-supervised approach of feature selection the first step followed is to pass a set of mandatory features to be used in every model and then obtain the models including mandatory ones and a best subset of the other features available.

The best subset of features is selected based on the highest predicted adjusted R^2 value [Reference 16] and Mallow's Cp [Reference 16]. The result of the best subset of feature selection is given below.

Best Subsets Regression: FE_COMPLEXITY versus family, Uncore_Reuse , ...

					Free Features							Mandatory Features		
Vars	R-Sq	R-Sq (adj)	Mallows Cp	S	family	uncore reuse	First gen PCI	Num new sockets	First gen memory	First gen other	new core	Family * Die combo	Family * Reuse	Core Count Die Size Ratio
1	87.3	83	12.5	15.87	X							X	X	X
1	85.2	80.3	15.7	17.08							X	X	X	X
2	90.4	86	9.8	14.41	X	X						X	X	X
2	90.3	85.9	9.9	14.45						X	X	X	X	X
3	94.5	91.2	5.5	11.44	X	X		X				X	X	X
3	94	90.4	6.2	11.95	X	X	X					X	X	X
4	**95.1**	**91.4**	**6.5**	11.31	**X**	**X**		**X**		**X**		**X**	**X**	**X**
4	95.1	91.2	6.6	11.4	X	X	X	X				X	X	X
5	95.7	91.3	7.7	11.34	X	X	X	X		X		X	X	X
5	95.3	90.5	8.3	11.85	X	X	X	X			X	X	X	X
6	96.1	91	9	11.53	X	X	X	X	X	X		X	X	X
6	95.7	90.1	9.6	12.11	X	X	X	X		X	X	X	X	X
7	96.1	89.6	11	12.43	X	X	X	X	X	X	X	X	X	X

Table 13: Illustrates the best Subsets Regression to select the best set of features from the free features.

Table 13 shows the predicted error estimations R^2 and adjusted R^2 for different subset selected from the available free features. The set which showed a maximum R^2adjusted and have almost equal Mallows Cp value and the number of features used is selected to build the model (highlighted).

The regression equation is

FE_COMPLEXITY (y) = - 73.60
+ 2.69 family
+ 0.92 Uncore_Reuse
+ 15.30 New_tech_sw_power_other
- 233 CC_Die_ratio
- 39.1 Num_new_sockets
- 0.03 Family*Reuse
- 0.1 Family*combo

In the above equation, the use of negative coefficients against variables CC_Die_Ratio, Num_New_Sockets and Family*Die_Package_Combo (highlighted by making bold) are against human intuition because as per experts a rise in those factors increases the complexity of the project (target).

One possible mathematical explanation is correlation among features. When two features are highly correlated, and both impact the target in the same direction, probably the effect of both of the features can be borne by any one of them. And the other one is used for adjustment of the residual. For Example, CC_Die_Ratio is highly correlated with family (correlation is .8) and impact of family on target is much higher in magnitude (which maintains intuitively correct sign +ve) than that of CC_Die_Ratio .

The impact of Family*Reuse is negative as per expectation. Though there is a term Uncore_Reuse with positive coefficient, overall impact of reuse features is negative as impact of Family*Reuse is higher in magnitude than the other one.

Goodness of Fit: S = 11.3110 R-Sq = 95.1% R-Sq(adj) = 91.4%

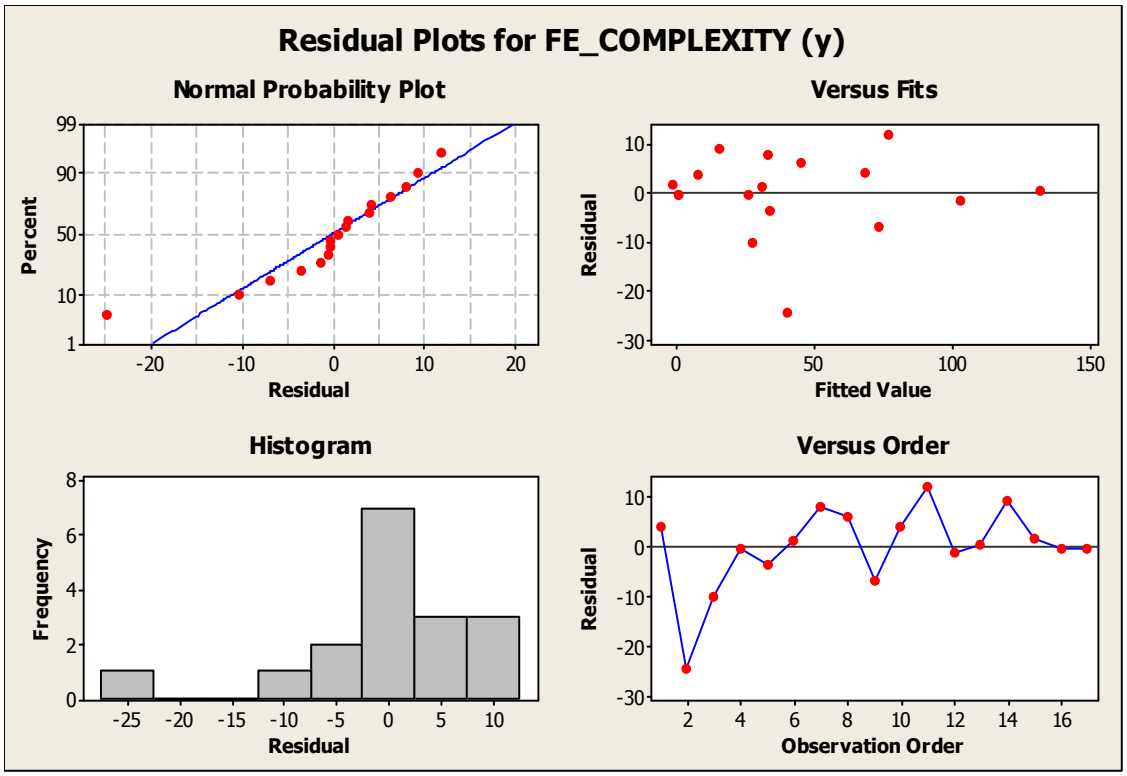


Figure 14: Illustrates the residual plots for model built using semi-supervised feature selection

Result shows with the mandatory use of some attributes the model's performance gave poor R^2 and R^2 adjusted. From the plot it is clearly visible that errors became higher as Maximum residual error=25. The histogram plot shows there is only one outlier for which the error is -25, for other data the error lies between -10 to +10. The data also has a non-linear nature.

To reduce the effect of non-linearity in response variables one possible solution is to use non-linear regression. The use of composite and polynomial terms and derived variables ensures better prediction of the target. But it is observed that the use of polynomial terms makes the system overfitted which results in a rise in test error for unknown data.

Keeping in mind that with these data samples there is not much scope to improve the model to fit everything properly, as an alternative it was decided to remove "the bicycles from the cars". Removing 4 projects with very low complexity from the training dataset and rebuilding the model made the model stable.

Changes done in data are given below in detail:

- Removal of the outlier data which showed -25 residual error in the last attempt. Removal of all data with the same product family of that data. (fortunately there was only one)
- Removal of 3 projects with very low complexity e.g. 1. The reason behind their removal include:
    - However small the residual error for them, error percentage is very high as their complexity value is very small.
    - They are highly leveraged project which kept front end design heads almost constant across different releases.

  So it is not wise to put those tiny pieces with giants, making the model's performance worse.
- There was one project for which there was no "core reuse" and that happened as an unexpected consequence of some organization problem. In future it is very unlikely that server project will not get core leverage from client. So the complexity due to core design was deducted from the overall complexity of the project. Too many binary features create difficulty for linear regression to pick the best subset, so one of the binary features is removed. Later, the effect of core is added to the model output equation.

6.4.3 Attempt 3: Semi-Supervised (with reduced dataset, by removing the odd ones)

Using the reduced set of training data the following steps are followed:

- Step 1: Best subset selection from the rest of the features while 3 features have been already passed as mandatory. Best subset is picked on the basis of adjusted R^2 and Mallow Cp number.
- Step 2: Stepwise feature selection based on increasing P value thresholds for enter and stay.
- Step 3: Use the features to build a new model.

Best Subsets Regression: FE_COMPLEXIT versus family, Uncore_Reuse, ...

					Free Features						Mandatory Features		
Vars	R-Sq	R-Sq (adj)	Mallows Cp	S	family	uncore reuse	First gen PCI	Num new sockets	First gen memory	First gen other	Family * Die combo	Family * Reuse	Core Count Die Size Ratio
1	89.2	88.8	32.3	11.3						X	X	X	X
1	89.1	88.7	32.6	11.3	X						X	X	X
2	95.3	92	14.4	7.9	X				X		X	X	X
2	94.4	90.3	17.5	8.7				X	X		X	X	X
3	97.6	95.1	9	6.2	X				X	X	X	X	X
3	96.1	92.3	13.7	7.8	X			X	X		X	X	X
4	98.8	97	7.1	4.8	X	X			X	X	X	X	X
4	98.4	96.1	8.3	5.5	X	X	X			X	X	X	X
5	99	96.9	8.4	4.9	X	X	X		X	X	X	X	X
5	98.9	96.8	8.5	5	X	X		X	X	X	X	X	X
6	99.1	96.3	10	5.39	X	X	X	X	X	X	X	X	X

Table 14: Illustrates the best Subsets Regression to select the best set of features from the free features after data set reduction.

Table 14 shows the list of subsets of features with predicted error estimations in terms of R^2 and adjusted R^2 and Mallows Cp value. The highlighted one indicates the combination selected. It gives the highest R^2 (adj) and comparable Mallows Cp with the number of features used.

Stepwise Regression: FE_COMPLEXITY (y) versus family, Uncore_Reuse, ...

Alpha-to-Enter: 0.09 Alpha-to-Remove: 0.09

Response is FE_COMPLEXITY (y) on 10 predictors, with N = 13

Step	1	2	3	4	5
Constant	22.30	-10.96	-16.94	-11.48	-23.41
CC_Die_ratio	611	480	452	453	472
T-Value	3.80	3.68	4.09	6.28	8.25
P-Value	0.004	0.006	0.005	0.001	0.000
Family*Reuse	-0.0030	0.0027	-0.0011	-0.0026	-0.0063
T-Value	-1.48	1.05	-0.39	-1.34	-2.77
P-Value	0.174	0.326	0.710	0.228	0.039
Family*combo	0.100	0.113	0.075	0.032	0.041
T-Value	3.18	4.64	2.77	1.42	2.27
P-Value	0.011	0.002	0.028	0.206	0.073
New_tech_sw_power_other	34.7	24.3	18.4	20.2	
T-Value	2.78	2.10	2.35	3.27	
P-Value	0.024	0.074	0.057	0.022	
family	0.40	0.48	0.59		
T-Value	2.08	3.74	5.24		
P-Value	0.076	0.010	0.003		
first_gen_memory_tech	29.0	25.6			
T-Value	3.23	3.56			
P-Value	0.018	0.016			
Uncore_Reuse	0.31				
T-Value	2.18				
P-Value	0.081				
S	15.0	11.3	9.52	6.22	4.88
R-Sq	78.82	89.22	93.33	97.56	98.75
R-Sq(adj)	71.77	83.83	88.57	95.13	97.00

Table 15: Illustrates stepwise Regression to select the best set of features from the free features after data point reduction. It selected the same set as selected in Table 14.

Stepwise selection shown in Table 15 also confirms the same set selected in best subset selection. The order in which they appear indicates their significance in predicting y (the algorithm is mentioned in Section 6.3)

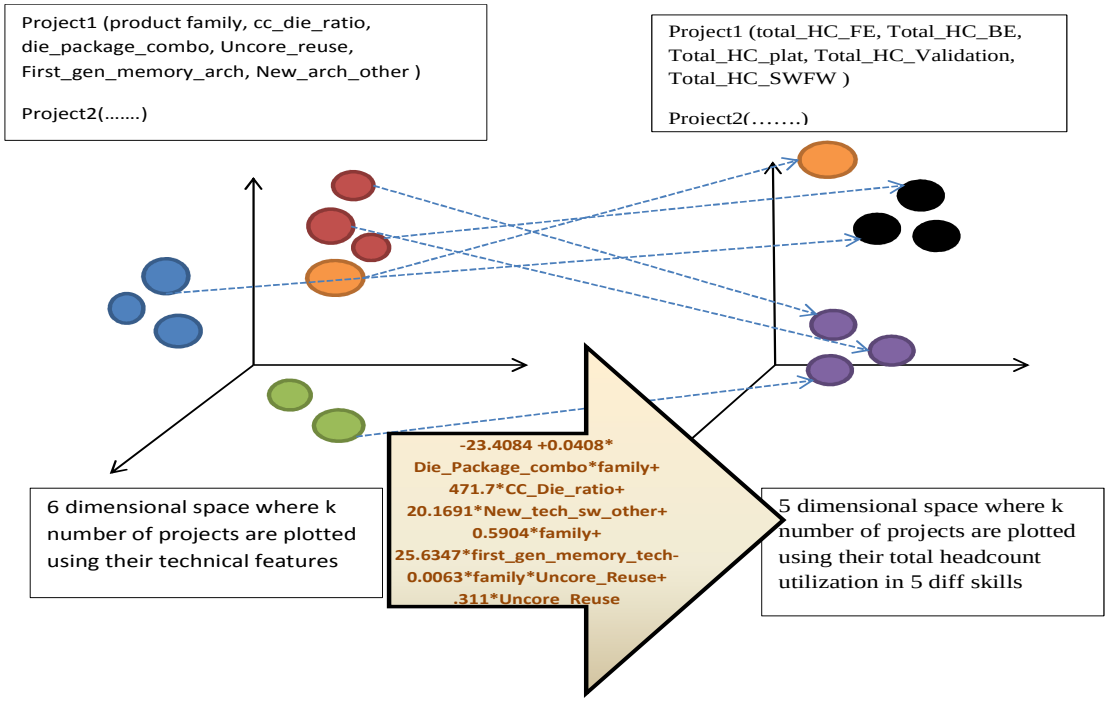


Figure 15: Illustrates the mapping/transformation function between the projects plotted in two different feature spaces.

FE_COMPLEXITY (y) = - 23.4084
+.0408* Die_Package_combo* Family
+ 471.7* CC_Die_Ratio
+20.1691*New_Tech_SW_Other
+.5904*Family
+ 25.635* First_Gen_Mem_tech
+ **(-.0063*Family + .311)*Uncore_Reuse**
+72*New_Core

In the above equation all of the features are bearing coefficient signs as per human intuition. For example, whenever Die package combo, CC_Die_Ratio increases or new core, new software technology, new memory tech comes into play that gives a significant rise in complexity increasing the headcounts. While Reuse is negatively correlated with target variable, it bears a –ve sign for its coefficient. Here is a small twist as we can see for smaller families if family*(-.0063) is less than .311 (highlighted by making bold in equation) the negative effect of reuse will not be there. So for products of low complexity the effect of Uncore_reuse is not

matching with intuition. But we have already removed most of the low end projects from our scope. So this model can be used easily to predict the medium and high end projects' complexity.

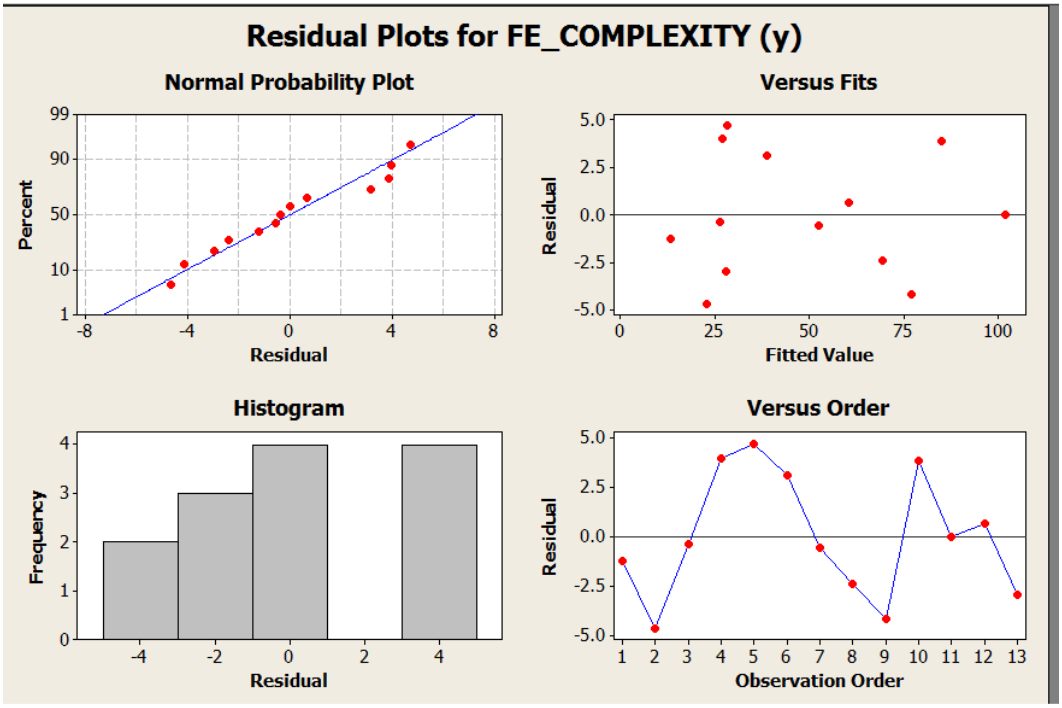


Figure 16: Illustrates the residual plots for model built using semi-supervised feature selection with data adjustment

Goodness of fit : R-Sq = 98.8% R-Sq(adj) = 97.0%

The improvements observed in the model trained with a reduced data set (removing the projects with very low complexity) are given below. –i) the non-linearity is reduced as the residual curve is not funnel type and ii) the range of errors is also reduced from 10 to 5. The normal probability plot for the residual is also improved with an improvement in R-Sq(adj).

6.4.4 Comments on Validation Method:

As the data set is very small, partitioning of the data into training set (60%) and validation set (40%) is not possible. But using 100% data for training and select the model based on least training error is not wise because this causes overfitting and generalization error or test error for future projects.

An alternative method is to apply cross validation and select the model showing least cross validation error. As the data set is small, if 20% of the data is taken out for validation and the

rest is used for training that affects the proper training of the regression model. So the 'leave one out' method is chosen for cross validation. Figure 17 taken from [Reference 8] shows the effect of overfitting.

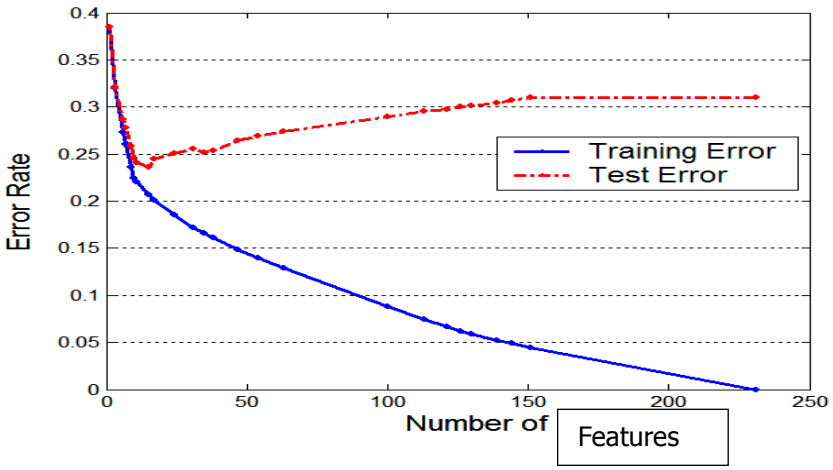


Figure 17: Illustrates the change is error rate as number of features increases

6.5 Comments on obtaining the headcounts for different skill bucket using the one complexity score

Front End Heads: As already discussed front end heads maintains linear relation with complexity score of the project. It is easy to obtain front end heads from complexity scores as shown in Figure 18 (a)

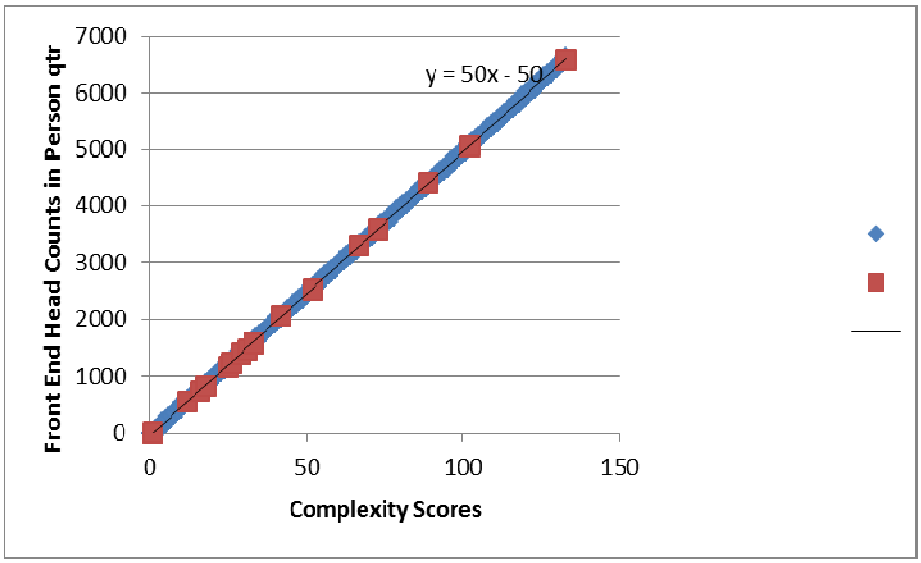


Figure 18(a) Relationship of front end heads requirement with complexity score

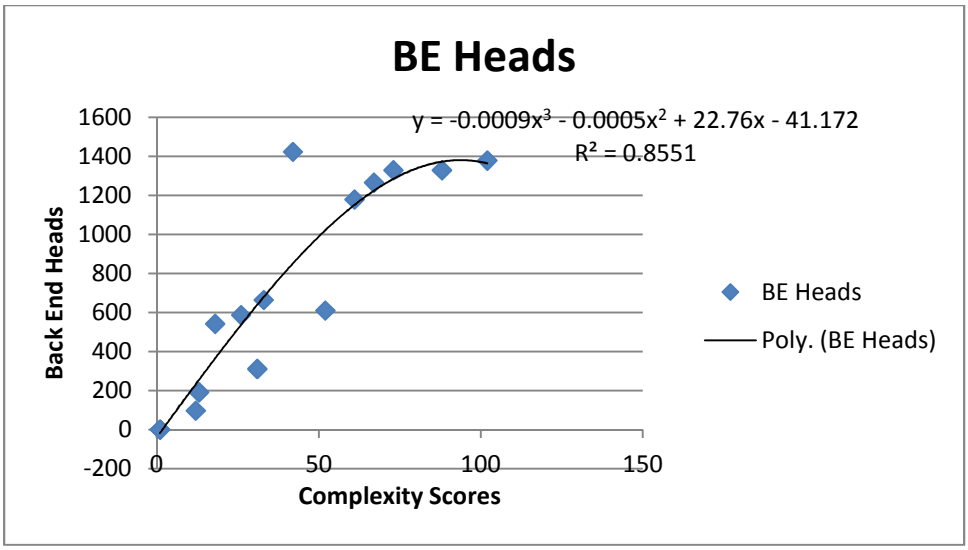


Figure 18(b) Relationship of back end heads requirement with complexity score

Back End Heads: The pattern shows Back end heads are somewhat constant at high complexity values. Only excluding 3 projects others got a good cubic fit, as shown in Figure 18(b).

Similarly, the other two skill buckets i.e. validation and software/firmware heads are also plotted against the complexity scores. Resource consumption for Validation engineers showed a similar pattern like resource consumption of backend heads for projects of higher complexity.

Platform heads: 80% of the projects showed almost constant platform heads across each product family. So platform heads will be predicted from the product family directly, there is no need to relate complexity scores etc. with platform heads.

Plat_HC_TOT		Product Family	Is predictable?
27		Family A	bad
69		Family A	bad
82		Family C	good
88		Family C	good
288		Family E	good
255		Family E	good
91		Family F	
8		Family G (A)	
136		Family G (B)	
104		Family H (A)	good
106		Family H (A)	good
738		Family H (B)	good
786		Family H (B)	good
44		Family H (C)	ok
65		Family H (C)	ok
295		Family H (C)	outlier

Table 16: Illustrates the consistency of platform heads across product families.

Table 16 shows platform engineering resources required by the projects in different product families. It is observed that except 3 projects (the projects belonging to Family A and one project in Family H(c)) platform resource requirements are very similar in projects belonging to a product family. So platform head requirement maintained consistency across product families. But to have more accurate prediction, similar regression models can be built to determine resource requirement for skill buckets other than FE.

Chapter 7

APPLYING PREDICTIVE MODEL ON UNKNOWN DATA

7.1 Testing Approach: Every model needs to be tested against some unknown data which are not used in training the model. In this thesis, the biggest challenge encountered in building and testing the model is lack of training examples. The initial plan was to set aside some project execution data for which both human plan and actual execution data were available for full project cycle. Those projects were not supposed to be used in training the model but instead supposed to be used as a test dataset. Because of too few training examples, no past project data could be used as test set. Fortunately, some ongoing projects were identified which were not used in training the model and have their 30-40% project execution data available in actuals.

As already mentioned, predictor model provides total heads in different skills required throughout the project. If the model output needs to be compared with 40% of actuals data available then obviously the total forecast have to be broken into quarterly resource demand. The possible ways include:

Approach 1:

Step 1: *Gather project duration information,*

Step 2: *Collect the total headcount demand predicted using the model,*

Step 3: *Collect Max headcount demand predicted using the model*

Step 4: *Plot a Gaussian over the duration having area equal to the predicted total headcount (obtained in step 2) and peak equal to max headcount (obtained in step 3).*

But in real life, project cycle patterns are not exactly Gaussian, rather a little skewed Gaussian towards the back. So if a simple Gaussian is plotted, the curves may have the same area but when actual and predicted are compared for a specific time slice (e.g. for each quarter or for a year), then results will not be satisfactory.

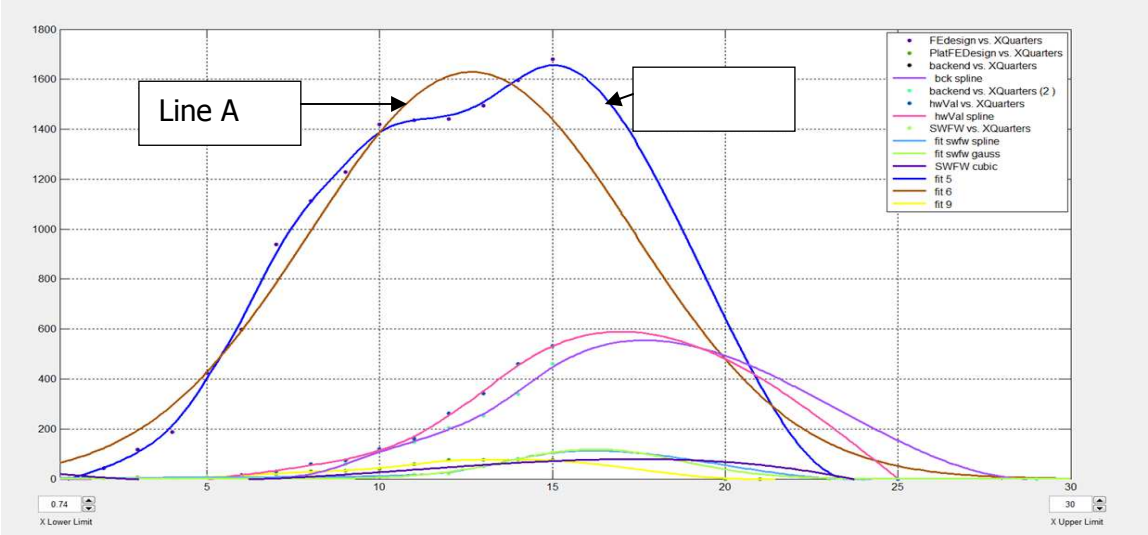


Figure 19: Illustrates two curves having same area can be different in each data point

Figure 19 confirms the fact that though the area under the curve (total heads) for actual (Line B) and Gaussian (Line A) curves are the same but when there is a matter of comparison for a specific duration (e.g. quarter wise) the error will go high. The error can occur in either positive or negative direction which ensures the increase in absolute error.

Approach 2:

Step 1: *Collect the duration of the future project from the product roadmap.*

Step 2: *Take actual data for the previous product belonging to same family.*

Step 3: *Compare the duration of both. If no match*

> ***Step 3.1:*** *Re-plot the actuals of the previous project having base equal to the base of the future project.*

Step 4: *Gather quarterly distribution (in %) of resource utilization from actuals by dividing the quarterly headcounts by total headcounts in the plot obtained in step 3.1.*

Multiply the distributions with the total headcount predicted by the model for the new project.

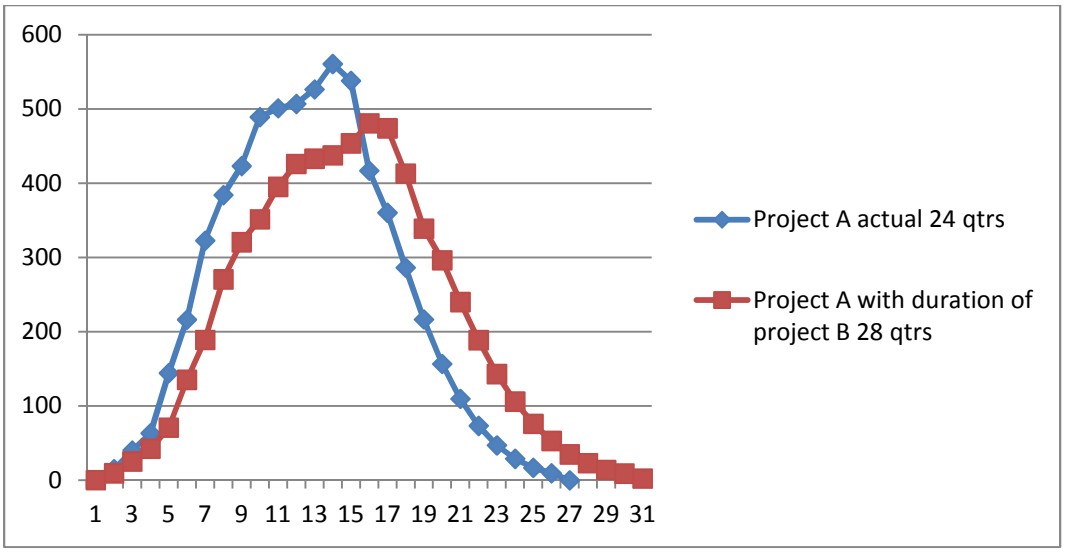


Figure 20: Resource Demand plotting with longer duration keeping total area under the curve same mentioned in step 3.1

Once the plot is done, we can find some timeframe in quarters for which both actual, model plan and human plan is available as described in the Figure 21. Here the comparison starts among those three using different error functions.

- Absolute error
- Signed error
- Max error
- Minimum error

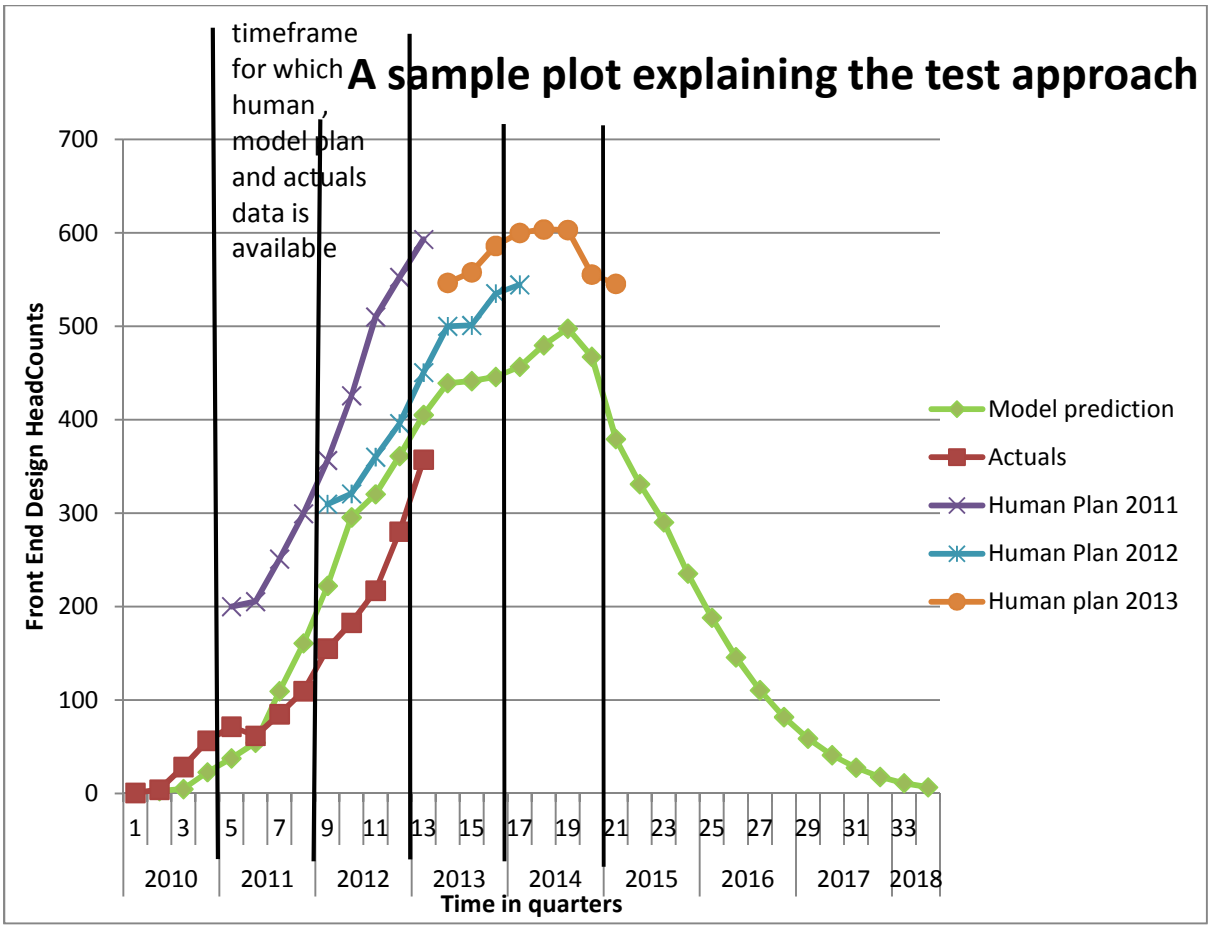


Figure 21: A Sample plot explaining the testing approach

As shown in Figure 21, the test strategy is explained below:

The best test scenario is to compare the model prediction with both the actuals and the human plans for the entire project cycle. If not possible, at least a comparison over longer duration is preferred. Model output is available for the entire project cycle. But we found that there are time period for which actuals are available (2010) and no human plans are present/maintained. Moreover, there are multiple versions of human plans released every year which contains the plan for next 2 years. Each version of the human plan differs significantly from past or future versions of it. So time is again sliced to have an accurate comparison between actual, model and different version of human plans.

Year 2010 – In year 2010, the only comparison possible is between Model predictions vs. Actuals as no human plan of that time is maintained.

- Model predictions vs. Actuals

Year 2011-12 – Model, Actuals and Human plan, all of the three are present for this time period. So both model forecast and human plan (2011 plan) can be compared against Actuals to determine which one is more accurate. Possible comparisons are:

- Model predictions vs. Actuals
- Human plan (2011 plan) vs. Actuals

Year 2012 – For the year 2012, model output is present, actuals are present and two versions (plan 11 and plan 12) of the human plan are present as every year human plans are released for next 2 years. So possible comparisons are

- Model prediction vs. Actuals
- Human plan (2011 plan) vs. Actuals
- Human plan (2012 plan) vs. Actuals

Year 2013 – For year 2013 no actuals are available as projects are yet not executed for 2013. So the only comparison possible is model to model comparison.

- Model predictions vs. Human plan (2012 plan)

Year 2013-14 – For 2013-14 as well no actuals are available but for year 2013 two versions of human plans are available. Human plan 2012 was compared in above segment. Now model prediction is to be compared with human plan 2013.

- Model predictions vs. Human plan (2013 plan)

7.2.1 Results: (Test Project 1)

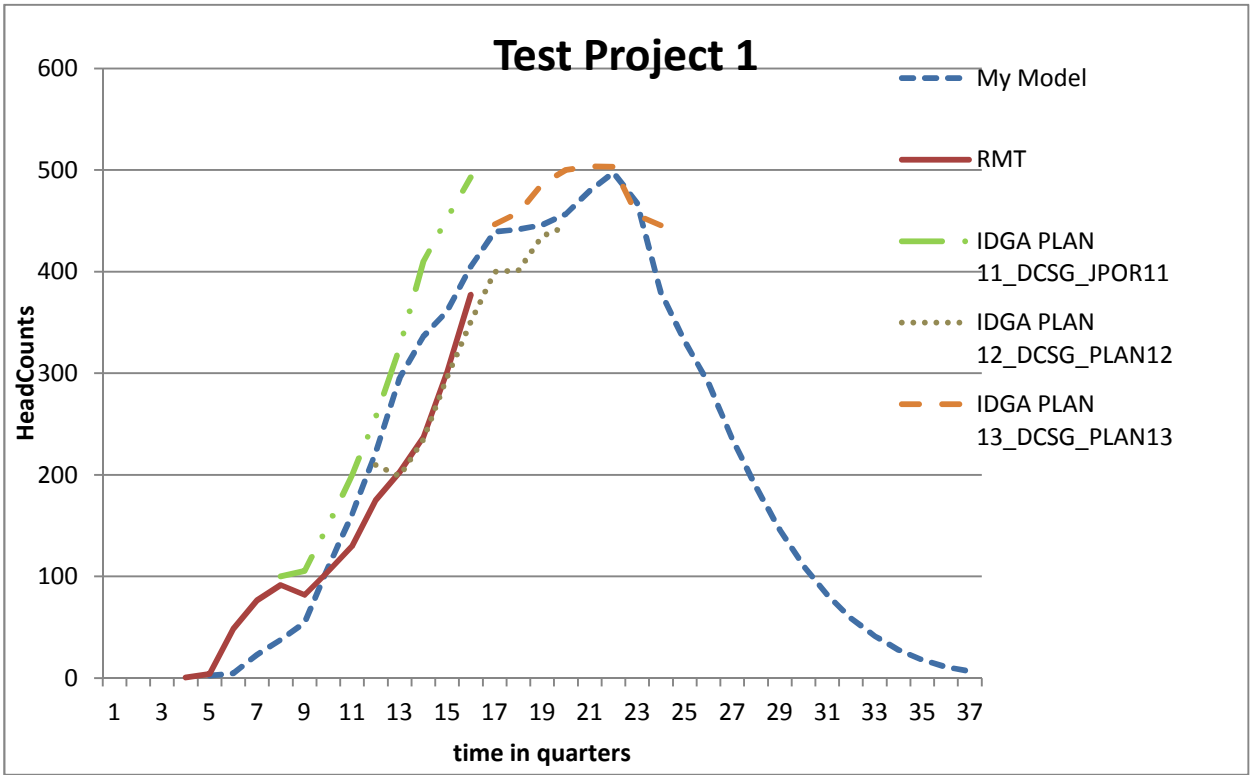


Figure 22(a): Model prediction, actuals and human plan comparison for project1

Errors	Duration	Model-Actual	Human plan 11-Actual	Human plan 12 - Actual	Model-Human Plan 12	Model-Human Plan 13
Absolute	2010-2012	542 (29%)				
	2011-2012	389 (22.8%)	783 (46%)			
	2012	279 (25%)	562 (50%)	38 (3%)		
	2013				103 (6%)	
	2013-2014					214 (5.6%)
Signed	2010-2012	181 (9.8%)				
	2011-2012	334 (20%)	783 (46%)			
	2012	279 (25%)	562 (50%)	-38 (-3%)		
	2013				103 (6%)	
	2013-2014					-191 (-5%)
Max	2010-2012	99				
	2011-2012	99	172			
	2012	99	172	-26		
	2013				40	
	2013-2014					-66
Min	2010-2012	4				
	2011-2012	4	8			
	2012	27	115	-2		
	2013				10	
	2013-2014					-5

Table 17 (a) : Model comparison with actuals and human plans

7.2.2 Results: (Test Project 2)

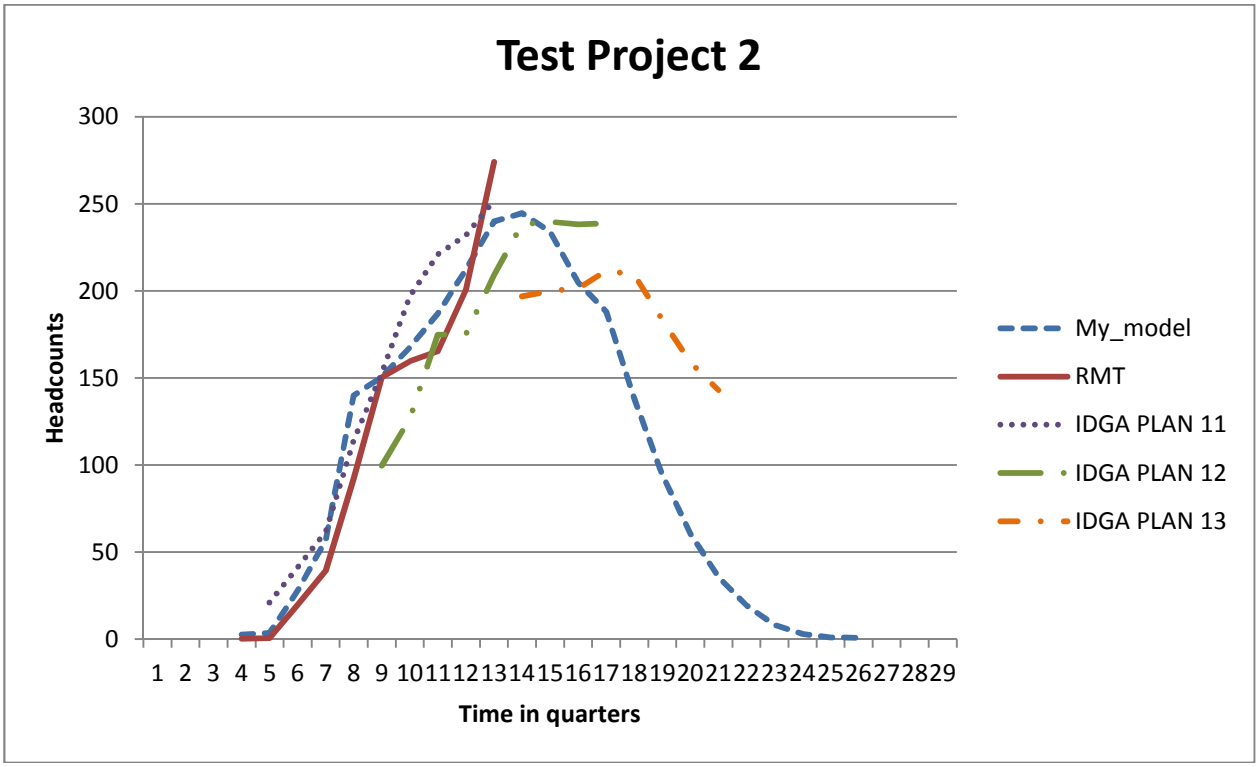


Figure 22(b): Model prediction, actuals and human plan comparison for project2

Errors	Duration	Model-Actual	Human plan 11-Actual	Human plan 12 - Actual	Model-Human Plan 12	Model-Human Plan 13
Abs	2011-2012	160(14.5%)	214 (19.5%)			
	2012	87 (10.9%)	144 (18.14%)	132 (16.5%)		
	2013				44 (4.6%)	
	2013-2014					340 (22.6%)
Signed	2011-2012	83 (7.5%)	170 (15.5%)			
	2012	-42(-5.25%)	102 (12.7%)	-113 (-14%)		
	2013				-31 (-3.2%)	
	2013-2014					-100 (-6.64%)
Max	2011-2012	-61	55			
	2012	-61	55	-65		
	2013				-34	
	2013-2014					-82
Min	2011-2012	2.22	2.43			
	2012	2.22	10.33	9.41		
	2013				1.67	
	2013-2014					-6.83

Table 17 (b): Model comparison with actuals and human plans

7.3 Analysis:

The model is applied on two unknown projects (test project 1 and test project 2) which were never used in training the model.

There are mainly two types of errors that are useful to analyze the accuracy of our model and human plans after comparison with project actuals: Absolute Error and Signed Error.

From Table 17(a) for test project 1, for timeframe year 2011-12, our model to actual comparison error is 22.8% (absolute error) and 20% (signed error), whereas initial human plan (plan 11) to actuals comparison error is 46% (both absolute and signed error) for the same time frame. This indicates that initial the human plan was overly pessimistic and the over-estimation trend was continued for 2 years in all quarters as absolute and signed error percentages are same and positive. Signed error for our model-actual comparison (20%) for 2011-12 is less than its absolute error (22.8%). So for some quarters our model did over-estimation and for some quarters it did under-estimation, over-estimation dominates though.

The plan for 2012 is taken from human plan released in 2011 and is compared with 2012 actuals. It showed even higher over-estimation of 50% (both signed error and absolute error are same here); while the model did 25% over-estimation for 2012.

Now in 2011 another plan for 2012 is released which contains the plans for the years 2012 and 2013. This time the human planners have more visibility of the project. The 2012 part of the new plan is compared with actuals for 2012 which showed 3% under-estimation while the model showed 25% over-estimation.

So for the test project1 Model output is better than initial human plan but worse than the next version of human plan. In human plan vs actuals comparison, the signed error is quite close to the absolute error for all durations. That means the trend of over-estimation or under-estimation is consistently maintained over quarters for human plan. The model is better than the human plan in terms of max and min error also.

From Table 17(b) the results of model-actuals and human-actuals comparison confirm the above fact again. Model output is slightly better than initial human plan 2011-12 in

terms of absolute error. But while comparing signed error, the model showed an overall under-estimation compared to actuals while the initial human plan showed over estimation.

For this project the surprise comes while comparing the model output and actuals with the second version of human plan. It is overly optimistic and for one year 2012 the prediction is -14% below than actual. This is not desirable because after one year of the beginning of project the plans should be more accurate as seen for test Project 1. So an important observation is the model did better than the human plan version 2 for test project 2.

It can be concluded that the pattern shows the human plans are too pessimistic in the beginning of a giant project and as a result of that over-estimation of resources occurs. When time passes they become too optimistic which results in under-estimation of the resources. Therefore, human plans are biased, whereas the model plan is free from bias as it learns from Actuals.

Another advantage of using the model for resource demand forecast is: Early availability of the resource demand for the complete project cycle right after the technical feature planning phase. Generally, human plans are generated every year for next 2 years whereas projects run for 5-6 years. It is beneficial to have a resource demand forecast for the complete cycle of the project at the beginning of project for better capacity planning and recruitment decisions. So this model ensures early availability of resource plan with negligible effort, whereas human plans need a considerable amount of heads to do the estimation.

Conclusion: When the model is trained with more data, the system learns better, giving rise to better predictions. This desired scenario is not very common in the real world. It is observed that the frequency of product release in Semiconductor industry is low. As a result it is found that not too many Semiconductor projects execution data is available to train the predictive system at any point of time. Moreover, considering the fact of changing world and technology and people the actuals for the projects executed 10 years back or 20 years back will not be of much use for future prediction. So the model has to always deal with 17-30 data points as training set which

are considered as "too few" in classical data mining. Here is the contradiction between classical concepts and real life scenario. And this thesis endeavors to figure out how to come up with a good predictive model with this constraint of "too small dataset".

Chapter 8

FUTURE WORK

Future extension of this work could be thought of in many different ways. Resource planning will result in improvement when applied under the following features.

Improvement in resource planning by providing the following features:

- Use the forecast at detail job role level and then aggregate them into high level groups for better understanding and for having an "easy to use" model. Moreover, when required users can drill down to detailed job roles etc. But that approach has its own challenges like too many skills (dimensions) and too few data.
- In addition to job roles, use resources' expertise levels / "job grade" as well in the output of the predictive model. The rationale behind this is in a good planning system the junior resource and a senior resource cannot be given equal importance. So the model prediction is going to be a "server project A" is going to take 100 "skill Z" 8-10 years experienced engineers, 500 "skill Z" 1-4 years experienced engineer , 600 "skill V" 2-4 years experienced engineers etc.
- Include site/work location information in planning by analyzing the impact of using resources from multiple work location (while sites are geographically very diverse) on project execution, performance, cost and time.

This business problem gives rise to many other critical business problems:

Once the resource demand for future projects is in place and availability of the resources in different divisions and geographic location are also available, there can be an optimization problem to figure out the best possible resource fit. How appropriate a resource for a project can be measured in terms of skill, job experience, site/location and cost?

This will help in

- feasibility analysis of future projects,

- deciding roadmaps,
- making long term/short term recruitment plans,
- better resource utilization across different divisions within an organization by sharing resources,
- site detection for future projects

In terms of methodology, other data mining methods can be applied e.g. ANN, SVM to check whether there is a possibility of getting a better model for demand forecast.

For demand and supply optimization problem integer programming, linear programming can be used.

REFERENCES

[1] Hu, J., Ray, B. K., & Singh, M. (2007). Statistical methods for automated generation of service engagement staffing plans. *IBM Journal of Research and Development*, *51*(3.4), 281-293.

[2] Hu, J., Singh, M., & Mojsilovic, A. (2008). Using data mining for accurate resource and skill demand forecasting in services engagements. In *Proc. KDD Workshop on Data Mining for Business Applications, Association for Computing Machinery, New York* (pp. 12-17).

[3] Datta, R., Hu, J., & Ray, B. (2008, June). Sequence mining for business analytics: Building project taxonomies for resource demand forecasting. In*Proceeding of the 2008 conference on Applications of Data Mining in E-Business and Finance* (pp. 133-141).

[4] Yoshimura, M., Fujimi, Y., Izui, K., & Nishiwaki, S. (2006). Decision-making support system for human resource allocation in product development projects.*International journal of production research*, *44*(5), 831-848

[5] Heimerl, C., & Kolisch, R. (2010). Scheduling and staffing multiple projects with a multi-skilled workforce. *OR spectrum*, *32*(2), 343-368.

[6] Dixit, K., Goyal, M., Gupta, P., Kambhatla, N., Lotlikar, R. M., Majumdar, D., ... & Soni, S. (2009, September). Effective decision support for workforce deployment service systems. In *Services Computing, 2009. SCC'09. IEEE International Conference on* (pp. 104-111). IEEE.

[7] Chenthamarakshan, V., Dixit, K., Gattani, M., Goyal, M., Gupta, P., Kambhatla, N., ... & Visweswariah, K. (2010). Effective decision support systems for workforce deployment. *IBM Journal of Research and Development*,*54*(6), 5-1

[8] Tan, P. N. (2007). *Introduction to data mining*. Pearson Education India.

[9] Berson, A., & Smith, S. J. (1997). *Data warehousing, data mining, and OLAP*. McGraw-Hill, Inc..

[10] Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin*, *23*(4), 3-13.

[11] Azevedo, A. I. R. L. (2008). KDD, SEMMA and CRISP-DM: a parallel overview.

[12] Steinbach, M., Karypis, G., & Kumar, V. (2000, August). A comparison of document clustering techniques. In *KDD workshop on text mining* (Vol. 400, pp. 525-526).

[13] Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, *66*(336), 846-850.

[14] Fayyad, U. M., 1996. Data mining and knowledge discovery: making sense out of data. IEEE Expert, Vol. 11 No. 5, pp20-25.

[15] Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78-87.

[16] Montgomery, D. C., Peck, E. A., & Vining, G. G. (2012). *Introduction to linear regression analysis* (Vol. 821). Wiley. Chapters 3,4,7,8

[17] Wikipedia. (2004). *Intel Tick-Tock*. Retrieved from http://en.wikipedia.org/wiki/Intel_Tick-Tock